

Desenvolvimento de um Dispensador Automático de Medicamentos

Trabalho de Projeto apresentado para a obtenção do grau de Mestre em
Automação e Comunicações em Sistemas de Energia

Autor

Nelson Miguel Domingues Vinagre

Orientadores

Doutor Inácio de Sousa Adelino da Fonseca

Doutor José Manuel Torres Farinha

Instituição

Instituto Superior de Engenharia de Coimbra

Coimbra, Dezembro, 2013

Agradecimentos

Quero deixar, aqui, expresso o meu reconhecimento:

- Aos meus orientadores de projeto, professor doutor Inácio Fonseca e professor doutor José Torres Farinha, disponibilidade, persistência, sabedoria e conselhos preciosos e por me encorajar a investigar e nunca desistir.
- Ao pessoal docente e não docente do Departamento de Engenharia Eletrotécnica pelo apoio e disponibilidade, e em particular ao professor Marco Silva pela ajuda na soldadura das PCI (Placas de Circuito Impresso) e ao Sr. Francisco Dias pela ajuda no fornecimento de material;
- À restante equipa de trabalho que me acolheram desde o primeiro momento e me impulsionaram para ter hoje este produto;
- Aos desenvolvedores e contribuidores de projetos com licenças permissivas (de software ou outros conteúdos) que pouparam muitas horas de trabalho neste projeto.
- Por fim, mas não menos importante quero agradecer à minha família e amigos pelo apoio prestado durante este longo percurso.

Resumo

O número de pessoas com idade superior a 65 anos aumentou consideravelmente nos últimos 40 anos. Este incremento da longevidade tem levado ao aparecimento de diversas patologias relacionadas com a idade e ao aumento da prevalência das patologias crónicas. Na grande maioria, esta população idosa é polimedicada, apresentando idades compreendidas entre os 60 e os 90 anos. Tomando como exemplo a *Diabetes Mellitus Tipo 2*, uma doença crónica onde, maioritariamente, os pacientes são polimedicados e com alguma idade, alguns estudos sugerem que 50% dos doentes não aderem aos regimes medicamentosos e que apenas um terço dos doentes diabéticos têm uma adesão adequada à terapêutica. Ora, a não-adesão à terapêutica, para além de ser um problema significativo para os doentes, uma vez que diminui a eficácia do tratamento e aumenta o risco de desenvolvimento de complicações, aumenta substancialmente os custos humanos, sociais e económicos da doença.

O presente projeto de mestrado foi materializado num dispositivo, designado *ePillbox*, que visou responder aos problemas da polimedicamentação, através de uma solução tecnológica, incorporando as vertentes de controlo e comunicação. Por outras palavras, o presente projeto pretende ser uma solução de engenharia para um problema de saúde que se prevê continuar a agravar-se nos próximos anos.

O dispositivo final é encapsulado numa caixa individual com um *design* inovador, e mais-valias tecnológicas, relativamente aos dispositivos existentes no mercado.

Como resultado final, o presente projeto de mestrado inclui não só uma vertente tecnológica, que está subjacente ao presente mestrado, mas também uma vertente de *design* e de serviço a um público que poderá ver na presente solução um grande valor acrescentado na sua qualidade de vida.

Palavras-chave: Dispensador de medicamentos; Medicamentos automáticos; Dispensador individual; Dispensador eletrónico de medicamentos

Abstract

The number of people aged over 65 has increased considerably in the last 40 years. This increase in longevity has led to the emergence of various diseases related to age and the increased prevalence of chronic pathologies. The vast majority of the elderly population, with ages between 60 and 90 years, is multi-medicated. For example, in the case of *Diabetes Mellitus*, some studies suggest that 50% of patients do not adhere to medication regimens and that only one third of them have adequate adhesion to therapy. However, non-adherence to therapy is a significant problem for patients, as it decreases the effectiveness of treatment and increases the risk of developing complications, also increasing the human, social and economic cost of disease.

This master's degree project presents a new device, called *ePillbox*, with the main object to solve the multi-medication problems, through a technological solution, with control and communications features built-in. In other words, this device aims to be an engineering solution to a health problem that is expected to get worse in the coming years.

The final device is packaged in an individual box with innovative design and technological improvements in relation to existing devices in the market.

As a final result, this master's project includes not only a technological aspect that underlies this master's degree, but also an element of design and service to an audience that can see in this solution a great added value in their quality of life .

Key-words: Pill dispenser; Automatic pill; Individual dispenser; Electronic pill dispenser.

Índice

Agradecimentos	i
Resumo	iii
Abstract.....	v
Índice	vii
Índice de figuras.....	xi
Índice de tabelas	xv
Abreviaturas	xvii
Capítulo 1 - Introdução	1
1.1 - Motivação e contexto	1
1.2 - O projeto e a equipa.....	2
1.3 - Objetivos	3
1.4 - Organização do documento	3
Capítulo 2 - Estado da Arte.....	5
Capítulo 3 - Arquitetura do sistema.....	9
3.1 - Enquadramento.....	9
3.2 - Arquitetura do sistema	9
3.3 - Módulos do dispositivo <i>ePillBox</i>	11
3.3.1 - Alimentação do sistema	12
3.3.2 - Unidade de processamento	12
3.3.3 - Módulos de comunicação	13
3.3.3.2 - Wi-Fi /Ethernet	14
3.3.3.3 - Bluetooth	14
3.3.4 - <i>Interface</i> com o utilizador	14
3.3.5 - <i>Interface</i> de potência com os atuadores	14
3.3.6 - Atuadores	15
3.3.7 - Tabuleiro de tomas	15
3.4 - Sumário.....	15
Capítulo 4 - Desenvolvimento do <i>hardware</i>.....	17
4.1 - Enquadramento.....	17
4.2 - Módulo de Alimentação.....	17
4.2.1 - A alimentação externa	18
4.2.2 - Conversor DC-DC	18
4.2.3 - Baterias	19

4.2.3.1 - A escolha do tipo de bateria	19
4.2.3.2 - Algoritmos de carga.....	21
4.2.4 - Circuito Carregador de Baterias	21
4.2.5 - Circuito de seleção de alimentação	22
4.3 - Módulo do microcontrolador	23
4.3.1 - O microcontrolador	23
4.3.1.1 - Pinagem do microcontrolador	23
4.3.1.2 - Pinos de alimentação do μ C	23
4.3.1.3 - Entradas Analógicas.....	24
4.3.1.4 - Circuito de Reset do μ C	27
4.3.1.5 - Interface de programação SWD (Serial Wire Debug).	27
4.3.1.6 - Opção de Arranque de sistema.....	28
4.3.1.7 - Fontes de relógio externo.....	29
4.3.2 - Chip de expansão de portas	29
4.3.3 - Slot de cartões <i>MicroSD</i>	31
4.3.4 - Porta <i>micro-USB</i>	32
4.3.5 - Ligação para módulo de alimentação	33
4.3.6 - Ligação para comunicação <i>Wi-Fi</i>	33
4.3.7 - Ligação para módulo de controlo de tampas	34
4.3.8 - Ligação para módulo de <i>interface Ethernet</i>	35
4.3.9 - Ligação para o módulo de comunicação <i>GSM</i>	36
4.3.10 - Ligação para uma mini-coluna de áudio	37
4.3.11 - Ligação para Entradas e Saídas Externas.....	37
4.3.12 - Ligação para o módulo de áudio	38
4.3.13 - Ligação para de comunicação <i>Bluetooth</i>	39
4.4 - Módulo de controlo dos atuadores das tampas	39
4.4.1 - Os atuadores das tampas	39
4.4.1.1 - O nitinol	40
4.4.1.2 - A aplicação	40
4.4.1.3 - Modelização	41
4.4.2 - Projeto e conceção do módulo	41
4.5 - Módulos de comunicação.....	43
4.5.1 - Módulo <i>Wi-Fi</i>	43
4.5.2 - <i>GSM</i>	45
4.5.2.1 - Adaptador de níveis de tensão.....	45
4.5.2.2 - Controlo de Alimentação e Reset.....	46
4.5.2.3 - Circuito do módulo.....	47
4.5.3 - <i>Bluetooth</i>	47
4.5.4 - <i>Ethernet</i>	48
4.6 - Desenho da placa de circuito impresso	49
4.6.1 - Normas e regras a respeitar.....	49
4.6.2 - Especificações técnicas	50
4.6.3 - Disposição de componentes	51
4.6.4 - Planos de massa	53

4.6.5 - Legenda da PCI	55
4.7 - Sumário	55
Capítulo 5 - Desenvolvimento de software	57
5.1 - Enquadramento	57
5.2 - O firmware	58
5.2.1 - O Sistema Operativo de Tempo Real (RTOS)	59
5.2.1.1 - Vantagens e desvantagens de um RTOS	59
5.2.1.2 - A escolha do RTOS	59
5.2.2 - Organização dos blocos de código	60
5.2.3 - Entradas e saída da <i>ePillBox</i>	61
5.2.4 - Descrição do modo de funcionamento do <i>firmware</i>	62
5.2.5 - Gestão de energia	65
5.3 - A <i>Interface</i> de Programação de Aplicativos (API)	65
5.3.1 - Comunicação <i>ePillBox</i> \leftrightarrow API	65
5.3.2 - O protocolo JSON-RPC	66
5.3.3 - Formato do objeto de pedido	66
5.3.3.1 - Formato do objeto de notificação	66
5.3.4 - Formato do objeto de resposta	67
5.3.4.1 - Formato do objeto "error"	67
5.3.5 - Listagem de procedimentos da API	68
5.4 - Sistema de Informação Web	68
5.4.1 - Introdução	68
5.4.2 - Ferramentas de desenvolvimento de sistema de informação <i>ePillbox</i>	70
5.4.3 - Informações sobre Web Sistema <i>ePillbox</i>	70
5.5 - Sumário	73
Capítulo 6 - Validação do dispositivo	74
6.1 - Testes efetuados ao <i>hardware</i>	74
6.1.1 - Produção das PCI	74
6.1.2 - Soldadura dos componentes	75
6.1.3 - O módulo da fonte de alimentação	76
6.1.4 - Alternativa ao módulo da fonte de alimentação	77
6.2 - Testes efetuados ao <i>Firmware</i>	77
6.2.1 - Teste do módulo do uC	77
6.2.2 - Teste do módulo de controlo dos atuadores das tampas	78
6.2.3 - Teste dos módulos de comunicação	79
6.2.3.1 - Teste do módulo Bluetooth	79
6.2.3.2 - Teste do módulo Ethernet	80
6.2.3.3 - Teste do módulo Wi-Fi	80
6.2.4 - Leitura e escrita no cartão <i>MicroSD</i> e conversão de sinais analógicos	81
Capítulo 7 - O design da <i>ePillbox</i>	82
7.1 - Enquadramento	82
7.2 - O exterior da <i>ePillbox</i>	83

7.3 - O interior da <i>ePillBox</i>	83
7.3.1 - Os compartimentos individuais	85
7.3.2 - Mecanismo de abertura das tampas dos compartimentos	86
7.4 - Sumário	86
Capítulo 8 - Conclusões	87
8.1 - Conclusões	87
8.2 - Trabalhos futuros.....	88
8.2.1 - Melhorias no <i>firmware</i> ;	88
8.2.1.1 - Sistema de Comunicações.....	88
8.2.1.2 - Geral	88
8.2.2 - Melhorias no <i>hardware</i>	88
8.2.3 - Melhorias no servidor remoto (páginas web e base de dados)	89
Referências bibliográficas	91
Anexos.....	97
Anexo A - Mapa da utilização dos pinos do microprocessador	
Anexo B - Esquemas eletrónicos do projeto	
B.1 - Interligação entre Módulos.....	B-1
B.2 - Esquema do módulo de atuação das tampas	B-2
B.3 - Esquema do módulo <i>Bluetooth</i>	B-3
B.4 - Esquema do módulo do μC	B-4
B.5 - Esquema do módulo GSM.....	B-6
B.6 - Esquema do módulo da fonte de alimentação	B-7
B.7 - Esquema do módulo <i>Ethernet</i>	B-8
B.8 - Esquema do módulo Wi-Fi.....	B-9
Anexo C - Desenho das Placas de Circuito Impresso	
C.1 - Placa do Módulo do μC	C-1
C.2 - Placa do Módulo de controlo das tampas	C-3
C.3 - Placa do Módulo da fonte de alimentação	C-4
C.4 - Placa do Módulo de bluetooth	C-5
Anexo D - Listagem e descrição das funções da API do servidor web	
D.1 - Listagem de procedimentos da API.....	D-1
Anexo E - Parametrização do módulo Wi-Fi (RN-171-XV WiFly)	
E.1 - Introdução	E-1
E.2 - Esquemático do módulo	E-1
E.3 - Listagem de parâmetros alterados	E-1
E.4 - Listagem de comandos para alterar os parâmetros	E-2
E.5 - Referências	E-3

Índice de figuras

Figura 2.1 – Alguns dispensadores automáticos de medicação circulares;	6
Figura 2.2 – Alguns dispensadores automáticos de medicação não circulares;	7
Figura 3.1 – Diagrama de blocos do sistema <i>ePillbox</i>	10
Figura 3.2 – Diagrama funcional da interação paciente, <i>ePillBox</i> e servidor remoto.....	11
Figura 3.3 – Diagrama de blocos da <i>ePillbox</i>	12
Figura 4.1 – Diagrama de blocos do módulo da fonte de alimentação.	17
Figura 4.2 – Características do CI LTC3536 [27, p. 4].	19
Figura 4.3 – Esquema do circuito do conversor DC-DC.	19
Figura 4.4 – Algoritmo de carga das baterias de <i>NiMH</i> (a) e <i>Li-Ion</i> (b), retirado de [30, pp. 3-4]	21
Figura 4.5 – Circuito de um carregador de 1A para uma célula de <i>Li-Ion</i> ou três células de <i>NiCd/NiMH</i> , retirado de [33, p. 10].	22
Figura 4.6 – Circuito do seletor de alimentação.....	22
Figura 4.7 – <i>Pinout</i> do microcontrolador STM32F407GVT6 [22, p. 40]	23
Figura 4.8 – Esquema de alimentação do μC	24
Figura 4.9- Esquema da ligação da porta <i>micro-USB</i> ao μC	25
Figura 4.10 – Esquema de Medição da tensão da bateria VBAT.	26
Figura 4.11 – Circuito de medição de corrente proveniente do módulo de controlo de Tampas	26
Figura 4.12 – Esquema de ligação do botão de reset ao μC	27
Figura 4.13 – Esquema da ficha SWD e interligação com o programador incluído na placa de desenvolvimento "STM32F4-Discovery";	28
Figura 4.14 – Esquema de ligação do pino Boot0 μC	29
Figura 4.15 – Circuito dos osciladores externos (a:HSE; b:LSE)	29
Figura 4.16 – Esquema do chip MCP23008	30
Figura 4.17 – Circuito de ligação do suporte do leitor de cartões <i>micro-SD</i> e o μC	32
Figura 4.18 – Circuito de ligação da ficha <i>Micro-USB</i> ao μC	33
Figura 4.19 – Circuito de ligação do conector ' <i>Power_CONN</i> '	33
Figura 4.20 – Circuito de ligação do conector ' <i>WiFi_CONN</i> '.	34
Figura 4.21 – Circuito de ligação do conector ' <i>Ctrl_Tampas_CONN</i> '.....	35
Figura 4.22 – Circuito de ligação do conector ' <i>Eth_CON</i> '.	36
Figura 4.23 – Circuito de ligação do conector ' <i>GSM_CON</i> '	37
Figura 4.24 – Circuito de ligação da mini-coluna presente no dispositivo.	37
Figura 4.25 – Circuito de ligação do conector ' <i>I/O_CON</i> '	38
Figura 4.26 – Circuito de ligação do conector ' <i>Audio_CON</i> '	39
Figura 4.27 – Circuito de ligação do conector ' <i>BT_CON</i> '.	39

Figura 4.28 – Alteração da estrutura cristalina do <i>nitinol</i> durante o processo de deformação, aquecimento e arrefecimento [44].	40
Figura 4.29 – Exemplo de aplicação do <i>nitinol</i> .	40
Figura 4.30 – Lista de características do <i>flexinol®</i> utilizado.	41
Figura 4.31 – Comparação entre os recursos necessários para controlo direto e controlo matricial.	42
Figura 4.32 – Esquema da matriz de controlo dos atuadores das portas dos compartimentos.	42
Figura 4.33 – Demonstração da passagem de corrente por diversas cargas quando estão ativas a linha e coluna 0 (no caso da ausência dos díodos de controlo do fluxo de corrente).	43
Figura 4.34 – Fotografia dos módulos “RN-XV WiFly”(A,B,C) e do “RN-171”(D).	44
Figura 4.35 – Circuito do módulo de comunicação com o Wi-Fi.	44
Figura 4.36 – Circuito de adaptação de níveis de sinal.	45
Figura 4.37 – Circuito de controlo de alimentação e <i>Reset</i> do Módulo MTSMC-G-F1.	46
Figura 4.38 – Circuito de controlo de alimentação e <i>Reset</i> do Módulo HC-06.	48
Figura 4.39 – Foto do Módulo BT_BOARD V1.4 e ligação à placa do módulo μC .	48
Figura 4.40 – a):Circuito de ligação do conector ‘Eth_CON’; b):Circuito de ligação do conector da placa “DP83848-Ethernet-Board” [50]; c): Imagem da “DP83848-Ethernet-Board”.	49
Figura 4.41 – Hierarquia das especificações de <i>design</i> (Serie 2220).	49
Figura 4.42 – Ecrã do <i>software</i> “PCB Calculator”	51
Figura 4.43 – Placa de circuito impresso do Módulo do μC .	52
Figura 4.44 – Placa de circuito impresso do Módulo da fonte de alimentação.	52
Figura 4.45 – Placa de circuito impresso do Módulo de controlo das tampas.	53
Figura 4.46 – Placa de circuito impresso do Módulo de Bluetooth.	53
Figura 4.47 – Planos de massa da PCI do Módulo do μC .	54
Figura 4.48 – Planos de massa da PCI do Módulo da fonte de alimentação.	54
Figura 4.49 – Planos de massa da PCI do Módulo de controlo das tampas.	54
Figura 4.50 – Planos de massa da PCI do Módulo de Bluetooth.	55
Figura 5.1 – Arquitetura do <i>software</i> do sistema.	57
Figura 5.2– Diagrama ilustrativo dos estados do <i>firmware</i> do dispositivo.	62
Figura 5.3 – Diagrama de temporal dos estados da <i>ePillBox</i> .	65
Figura 5.4 – Exemplo de uma troca de mensagens entre a <i>ePillBox</i> e o servidor Web.	66
Figura 5.5 – Diagrama das tabelas Mysql usando a ferramenta MySQL Workbench – Tabelas normalizada na terceira forma normal;	71
Figura 5.6 – <i>ePillBox</i> Sistema Web Site – frontend principal.	72
Figura 5.7 – <i>ePillBox</i> Sistema Web Site – inscrições novo utilizador;	72
Figura 5.8 – <i>ePillBox</i> Sistema Web Site – Medicação Schedule.	73
Figura 6.1– Fotografia das PCI do módulo do μC .	74
Figura 6.2 – Fotografia da PCI do módulo do μC parcialmente soldada (imagem da esquerda) e completamente soldadas (Imagens do centro e direita).	75

Figura 6.3 – Fotografia da PCI dos módulos da fonte de controlo das tampas e da fonte de alimentação.....	75
Figura 6.4 – Ampliação de algumas soldaduras para verificação da qualidade das mesmas.	76
Figura 6.5 – Forma de onda obtida na saída da fonte de alimentação.	76
Figura 6.6 – Circuito simulado a) e resultado da simulação b) no software LT-Spice IV.	77
Figura 6.7 – Imagem do módulo S7V8F3 a) e imagem do módulo montado numa placa de ensaio b).....	77
Figura 6.8 – Matriz de LED para simulação dos atuadores das tampas.	78
Figura 6.9 – Sequência de imagens que ilustram a atuação das tampas.	79
Figura 6.10 – Fotografia do módulo Bluetooth ligado ao μC e a imagem do programa XXX a mostrar os comandos enviados ao μC e as respostas enviadas de volta para o PC/tablet.	79
Figura 6.11 – Imagem da página Web e o estado do LED com conformidade com o que foi especificado na página.	80
Figura 6.12 – Demonstração do teste de entrada em modo de configuração, configuração através de uma página web e terminar as configurações.	80
Figura 6.13 – Demonstração da comunicação com o servidor web através do módulo Wi-Fi.....	81
Figura 7.1 – Imagem do conceito do dispositivo proposto.....	82
Figura 7.2 – Vista em prestativa da caixa fechada (vista frontal e traseira).	83
Figura 7.3 – Vista da caixa aberta com detalhes do compartimento do tabuleiro.	84
Figura 7.4 – Vista com detalhes sobre o interior da secção inferior do dispositivo.	85
Figura 7.5 – Pormenor do compartimento individual de tomas e do mecanismo de abertura da tampa.	85
Figura E.1 – (a)Imagem do módulo RN-171-XV; (b)Esquema de corresp. entre o número do pino e o nome[1];	E-1
Figura E.2 – Listagem de parâmetros alterados	E-1

Índice de tabelas

Tabela 3.1– Resumo das características do STM32F407. [22, p. 1].....	13
Tabela 4.1 – Comparação entre os 4 tipos de tecnologias de fontes de alimentação, adaptado de [26, p. 5].....	18
Tabela 4.2 – Tabela com a análise comparativa de alguns tipos de baterias, adaptado de [30, p. 1]..	20
Tabela 4.3 – lista de pinos analógicos utilizados.	24
Tabela 4.4 – Valores mínimos e máximos de tensão das baterias segundo o tipo de tecnologia utilizada no seu fabrico.	26
Tabela 4.5 – Listagem dos pinos que contêm o interface SWD.....	27
Tabela 4.6 – Modos de arranque	28
Tabela 4.7 – Lista de funções das E/S do CI MCP23008 para e respetiva localização nos conetores	30
Tabela 4.8 – Padrão de cartões MicoSD.	31
Tabela 4.9 – Lista das Correspondência entre o interface SDIO e o μ C.....	31
Tabela 4.10 – Lista das Correspondência entre o interface USB e o μ C	32
Tabela 4.11 – Lista das Correspondências entre o interface ‘Power_Conn’ e os diversos CI do Módulo	33
Tabela 4.12 – Lista das Correspondências entre o interface ‘WIFLY_Conn’ e os diversos CI do Módulo.	34
Tabela 4.13 – Lista das Correspondências entre o interface ‘Ctrl_tampas_Conn’ e os diversos CI do Módulo.	35
Tabela 4.14 – Lista das Correspondências entre o interface ‘ETH_Conn’ e os diversos CI do Módulo.	36
Tabela 4.15 – Lista das Correspondências entre o interface ‘GSM_Conn’ e os diversos CI do Módulo.	36
Tabela 4.16 – Lista das Correspondência entre o interface ‘I/O_Conn’ e os diversos CI do Módulo ...	38
Tabela 4.17 – Lista das Correspondências entre o interface ‘Audio_Conn’ e os diversos CI do Módulo.	38
Tabela 4.18 – Lista das Correspondência entre o interface BT_Conn’ e os diversos CI do Módulo	39
Tabela 4.19 – Especificações das pistas e vias.....	51
Tabela 5.1 – Lista de alguns tipos de objetos utilizados no protocolo JSON-RPC.	67
Tabela 6.1 – Valores obtidos do ficheiro de leituras das amostras das entradas analógicas.	81

Abreviaturas

AC – “Alternate Current”	GND – “Ground”
ADC – “Analog Digital Converter”	GNU – “GNU Is Not Unix”
ANSI – “American National Standards Institute”	GPIO – “General Propose Input Output”
API – “Application Programming Interface”	GPL – “GNU General Public License”
ARM – “Advanced RISC Machines”	GPRS – “General packet radio service”
ARP – “Address Resolution Protocol”	GSM – “Glogal System for Mobile Communications”
ASCII – “American Standard Code for Information Interchange”	HSE – “High Speed External”
AT – “Hayes command set”	HTML – “Hyper Text Markup Language”
BD – “Base de Dados”	I²C – “Inter-Integrated Circuit”
Bkp – “Backup”	I2S – “Inter-IC Sound ou Integrated Interchip Sound”
BPMN – “Business Process Model and Notation”	ICMP – “Internet Control Message Protocol”
CI – “Circuito Integrado”	IDE – “Integrated Development Environment”
CMSIS – “Cortex Microcontroller Software Interface Standard”	IEC – “International Electrotechnical Commission”
CPU – “Central Processing Unit”	IEEE – “Institute of Electrical and Electronics Engineers”
CRC – “Cyclic redundancy check”	IFML – “Interaction Flow Modeling Language”
CTS – “Clear To Send”	INT – “Interruption”
DAC – “Digital Analog Converter”	IP – “Internet Protocol”
DB – “Data Base”	ISEC – “Instituto Superior de Engenharia de Coimbra”
DC – “Direct Current”	JIT – “Just-in-Time”
DHCP – “Dynamic Host Configuration Protocol”	JSON – “JavaScript Object Notation”
DMU – “Dispensadores de Medicamentos por Unidose”	LED – “Light Emitter Diode”
DNS – “Domain Name Server”	Li-Ion – “Lithium Ion”
DOM – “Document Object Model”	LQFP – “Low Profile Quad Flat Package”
EDA – “Electronic Design Automation”	LSE – “Low Speed External”
EDR – “Enhanced Data Rate”	LTE – “Long Term Evolution”
EEPROM – “Electronic Erasable Programmable Read-Only Memory”	MCU – “Micro Controller Unit”
FAT – “File Allocation Table”	MII – “Media Independent Interface”
FTP – “File Transfer Protocol”	MISO – “Master Input Slave Output”
GDB – “The GNU Project Debugger”	MMCX – “Micro-Miniature Coaxial”
	MOSFET – “Metal–Oxide–Semiconductor Field-Effect Transistor”

MOSI – “Master Output Slave Input”	SDIO – “Secure Digital Input Output”
MVC – “Model View Controller”	SIM – “Subscriber Identity Module”
Ni-Cd – “Nickel-Cadmium”	SLA – “Sealed Lead Acid”
NiMH – “Nickel-Metal Hydride”	SMD – “Surface Mount Device”
Nss – “chip select”	SMS – “Short Message Service”
OEM – “Original Equipment Manufacturers”	SO – “Sistema Operativo”
OMG – “Object Management Group”	SPI – “Serial Peripheral Interface”
ORM – “Object-relational mapping”	SQL – “Structured Query Language”
OS – “Operating System”	SSID – “Service set identification”
OSI – “Open Systems Interconnection”	SWD – “Serial Wire Debug”
PC – “Personal Computer”	TCP – “Transmission Control Protocol”
PCB – “Printed Circuit Board”	Tx – “Transmit”
PECL – “PHP Extension Community Library”	UART – “Universal Asynchronous Receiver/Transmitter”
PHY – “Physical Layer”	UDP – “User Datagram Protocol”
PROM – “Programmable Read-Only Memory”	UML – “Unified Modeling Language”
PSK – “Pre-Shared Key”	UMTS – “Universal Mobile Telecommunications System”
PWM – “Pulse-Width Modulation”	US – “United States”
RAM – “Random Access Memory”	USART – “Universal Synchronous/Asynchronous Receiver/Transmitter”
REST – “Representational State Transfer”	USD – “United States Dollar”
RF – “Radio Frequency”	WADL – “Web Application Description Language”
RISC – “Reduced Instruction Set Computing”	WEB – “World Wide Web”
RMII – “Reduced Media Independent Interface”	WEP – “Wired Equivalent Privacy”
RPC – “Remote Call Procedure”	WKUP – “Wakeup”
RTCC – “Real Time Clock Calendar”	WPA2 – “Wi-Fi Protected Access II”
RTOS – “Real-Time Operating System”	WPA-PSK – “Wi-Fi Protected Access”
RTS – “Ready To Send”	WPS – “Wi-Fi Protected Setup”
Rx – “Receive”	WSDL – “Web Service Description Language”
SCL – “Serial Clock”	
SD – “Secure Digital”	
SDA – “Serial Data”	

Capítulo 1 - Introdução

1.1 - Motivação e contexto

A expectativa de vida na Europa Ocidental tem aumentado continuamente, desde 1950, em cerca de 2,5 anos por década, e espera-se que continue a aumentar, sendo esta, logo à nascença, uma clara tendência na Europa, [1, p. 2].

Em Portugal, a proporção de pessoas com 65 ou mais anos duplicou nos últimos 50 anos, passando de 8% em 1960, para 11% em 1981, 14% em 1991, 16,5% em 2001 [2, p. 169] e 18,2% em dezembro de 2010, [3, p. 73]. O Instituto Nacional de Estatística estimava que esta proporção duplicará nos próximos 50 anos, representando, em 2060, 40% da população portuguesa, na maior parte do território, [4, p. 7].

Recentemente, os Censos 2011 mostraram que o índice de envelhecimento em Portugal aumentou consideravelmente entre 1960 e 2011, respetivamente, 27,3% e 127,8%, [5]. Isto deve-se a vários fatores sociodemográficos que se traduziram num aumento da esperança média de vida, o qual tem levado ao aparecimento de diversas patologias relacionadas com a idade, bem como ao aumento da prevalência das patologias crónicas, tratadas sobretudo com medicamentos [6] e [7]. Na grande maioria, esta população idosa é polimedicada, ou seja, cada indivíduo consome cinco ou mais medicamentos. Maioritariamente, estes idosos apresentam baixos graus de literacia em saúde, o que limita o conhecimento e a compreensão da sua saúde e doenças, reduzindo a sua autonomia no autocuidado, em especial no cumprimento dos regimes posológicos dos medicamentos, [8]. Tomando como exemplo a *Diabetes Mellitus Tipo 2*, uma doença crónica onde, maioritariamente, os pacientes são polimedicados e com alguma idade, alguns estudos sugerem que 50% dos doentes não aderem aos regimes medicamentosos e que apenas um terço dos doentes diabéticos têm uma adesão adequada à terapêutica, [9] e [10]. Ora, a não-adesão à terapêutica, para além de ser um problema significativo para os doentes, uma vez que diminui a eficácia do tratamento e aumenta o risco de desenvolvimento de complicações, aumenta substancialmente os custos humanos, sociais e económicos da doença. Perante estes factos e numa conjuntura económico-financeira desfavorável, é urgente otimizar a gestão e o controlo da doença ao nível da medicação, particularmente nos doentes crónicos e idosos. Assim, ao nível dos cuidados primários de saúde, lares, centros de dia, entre outros, é possível criar dispositivos tecnológicos, tal como os dispensadores eletrónicos de medicamentos (*Pillbox*), capazes de organizar, gerir e facilitar a toma dos medicamentos.

Os DMU (Dispensadores de Medicamentos por Unidose) surgiram na década de 60 como um mecanismo eficaz para a diminuição dos erros na sua preparação e administração [11]. Neste âmbito, estes dispositivos responderam às exigências de que os medicamentos de cada paciente estivessem disponíveis na enfermaria a determinada hora e, portanto, necessitando de várias reposições diárias.

Assim, apenas uma pequena parte do tratamento do paciente estava disponível na unidade de internamento, pelo que todas as mudanças na prescrição ou no seu rastreamento ficavam refletidas na sua história fármaco-terapêutica e, adicionalmente facilitava a sua recompilação num curto espaço de tempo.

Nesta perspetiva, os critérios de um sistema de distribuição de medicamentos devem dar as seguintes respostas:

- Continuidade no fornecimento farmacêutico, objetivada pela validação prévia de todas as prescrições antes da dispensa dos medicamentos;
- Generalização, através da integração de toda a terapia necessária ao paciente e, portanto, de toda a gama de formas sólidas orais, líquidas, misturas intravenosas, até à fluidoterapia individualizada de substituição e de manutenção.

Passados mais de 40 anos desde a sua implementação, o conceito de DMU é uma realidade em cerca de 90% dos hospitais dos Estados Unidos, incluindo a terapia intravenosa, [12].

Este mesmo conceito, desde há vários anos, passou a ser aplicado a dispensadores individuais, existindo vários tipos de dispensadores automáticos, mais ou menos complexos, para diversos dias de uso, e com uma grande diversidade de preços. As figura 2.1 e figura 2.2 ilustram vários tipos de dispensadores individuais de medicamentos.

Estes dispensadores funcionam, quer ligados à corrente elétrica quer a baterias recarregáveis ou a pilhas, sendo programados com os horários nos quais deverá ser dispensada a medicação.

Os medicamentos são carregados no dispensador para um determinado período (o número de dias para os quais se pode preparar a medicação depende da capacidade do dispensador), sendo os mais vulgares para 18, 28, 40 ou mais tomas.

Os medicamentos para um dado momento são libertados na hora previamente programada, acompanhados de um sinal sonoro, visual ou outro (incluindo mensagem de voz). Usualmente, o alarme desliga-se quando o medicamento é retirado. Este tipo de solução evita a sobredosagem e diminui significativamente o incumprimento da toma.

Por consequência, os dispensadores automáticos de medicamentos facilitam o cumprimento dos horários de toma da medicação, especialmente para idosos com falhas de memória ou quando o cuidador tem mais do que uma pessoa a seu cargo. Adicionalmente, este tipo de dispositivo alivia os familiares, particularmente quando estes necessitam de se ausentar. Desta forma, a família sabe sempre quando o medicamento foi libertado, ou se ficou no dispensador. Este tipo de dispositivo é, ainda, geralmente protegido com um sistema de segurança que impede a toma inadequada de medicamentos, pois impossibilita a abertura dos contentores que não sejam os programados para a hora devida.

Contudo, estes dispositivos podem ser de difícil compreensão para alguns idosos, não sendo utilizáveis por pessoas com alterações cognitivas significativas. Para além destes aspetos, importa ainda atender às dificuldades inerentes à recarga dos contentores do dispensador.

1.2 - O projeto e a equipa

A ideia deste projeto surgiu do professor Rui Cruz, da Escola Superior de Tecnologias da Saúde de Coimbra (ESTeS). Visto ser um projeto que necessita de várias valências para ser concluído com sucesso constituiu-se uma equipa multidisciplinar, de várias instituições, que se passa a descrever:

- Escola Superior de Tecnologias da Saúde de Coimbra (ESTeS);
 - Professor Rui Cruz;
 - Professor Nelson Sá;
 - Professor Fernando Mendes;
- Instituto Superior de Engenharia de Coimbra (ISEC);
 - Professor Inácio Fonseca
 - Professor José Torres Farinha;
 - Nelson Vinagre;
- Escola Superior de Ensino de Coimbra (ESEC);
 - Professora Amparo Carvas;
 - Rui Dias (aluno finalista do Curso de Design Multimédia);
- FabLab – NovoTecná, Coimbra;
 - Eng. José Viana.

Cada instituição ficou responsável por uma parte do projeto, sendo que a parte que coube ao ISEC foi a do desenvolvimento da parte eletrónica, *firmware*, e página *web*.

1.3 - Objetivos

O principal objetivo do projeto de mestrado corresponde ao desenvolvimento de um sistema microcontrolado capaz de gerir um equipamento dispensador de medicamentos. O projeto consiste no desenvolvimento de *hardware*, *firmware* e uma página *web* para gestão desse mesmo equipamento.

Inicialmente foi necessário realizar uma pesquisa de mercado para identificar os equipamentos existentes que dão respostas ao problema em análise e à determinação das suas principais características. Posteriormente procedeu-se à análise de mercado referente à tecnologia necessária para a implementação do dispositivo, designadamente na vertente da Engenharia: microcontroladores, memórias, sensores, atuadores, baterias, módulos OEM para conectividade, etc.

Com esse trabalho elaborado procedeu-se ao desenvolvimento do circuito impresso (PCI) e da programação do equipamento para que pudesse cumprir os objetivos proposto. Foi ainda necessário proceder à implementação de um mecanismo de comunicação que possibilitasse a realização da parametrização, carregamento dos horários das tomas, e a verificação da evolução da toma dos medicamentos de forma remota, entre outras funcionalidades.

Seguiu-se a elaboração de um *sítio* na internet que permitisse a gestão centralizada da informação associada à utilização simultânea de vários equipamentos de dispensa de medicamentos.

Finalmente, o projeto foi terminado com a elaboração de toda a documentação referente ao projeto.

1.4 - Organização do documento

Esta monografia está dividida em nove capítulos, tal como se sintetiza seguidamente:

- O primeiro capítulo contém a introdução à monografia, a contextualização, os objetivos, as metas a atingir e a organização do documento.
- O segundo capítulo, contém o estado da arte relativamente os dispensadores individuais de medicamentos existentes no mercado internacional;
- O terceiro capítulo contém a arquitetura do sistema desenvolvido, sendo apresentado, em alto nível, o funcionamento do sistema;
- O quarto capítulo faz uma apresentação aprofundada do sistema, incluindo os diversos módulos que o constituem, bem como o *hardware* desenvolvido;
- O quinto capítulo descreve o *software* necessário ao funcionamento do sistema, e inclui ainda a referência ao *firmware*, à aplicação *web* e à base de dados;
- O sexto capítulo apresenta a validação do funcionamento do sistema, bem como a descrição dos testes e resultados obtidos;
- O sétimo capítulo descreve o *design* do dispositivo, tomando como referência o relatório de estágio do *designer*;
- O oitavo capítulo apresenta as conclusões, bem como eventuais trabalhos a desenvolver no futuro;
- O final da monografia é constituído pelas referências bibliográficas e os vários anexos mencionados ao longo dos capítulos precedentes.

Capítulo 2 - Estado da Arte

Desde sempre a humanidade tem aproveitado os recursos existentes para melhorar a sua qualidade de vida. É disso exemplo a grande diversidade de dispositivos, mais ou menos evoluídos tecnologicamente, que povoam as nossas vidas e nos auxiliam. A área da medicina não é exceção e, na vertente em estudo, já existem equipamentos que auxiliam as pessoas no cumprimento do seu regime de medicação regular.

O princípio de funcionamento destes dispositivos é semelhante, baseando-se no seguinte procedimento: o dispositivo tem um tabuleiro ou algo similar onde se colocam os medicamentos divididos por tomas; seguidamente configura-se com os horários das tomas; posteriormente é fechado, ficando pronto a iniciar a sua função; quando chega o instante programado fica disponível uma nova toma e o dispositivo avisa o utilizador; no caso de este não tomar o medicamento dentro do espaço de tempo previamente estabelecido, poderá ficar impedido de o fazer. Quase todos os dispositivos disponíveis no mercado são alimentados através da rede elétrica; no entanto, caso esta falhe, possuem uma bateria que mantém o dispositivo a funcionar num determinado intervalo de tempo.

Há equipamentos com funcionalidades adicionais. Os modelos mais dispendiosos possuem opções mais avançadas, tais como, por exemplo: monitorização e programação remotas; possibilidade de aviso do utilizador por telefone ou SMS; capacidade de alertar uma terceira pessoa, por telefone, SMS ou por correio eletrónico; e disponibilização de tabuleiros com a medicação pré-carregada.

A maior parte dos dispositivos existentes no mercado são cilíndricos, semelhantes a pratos. Possuem um tabuleiro onde são colocadas as tomas, encontrando-se dividido radialmente em compartimentos, tipicamente múltiplos de sete. Para condicionar o acesso aos medicamentos, existe uma tampa com uma pequena ranhura que possibilita o acesso somente à toma atual. Há alguns modelos onde esta tampa é trancada com recurso a uma chave.

Os dispositivos com o formato de prato existentes no mercado são os seguintes:

- **Med-e-lert** – Este equipamento fica-se pelas funcionalidades básicas, o que faz dele o mais simples e barato do conjunto – 69,95 USD, [13]. Tem capacidade para 28 tomas com até quatro alarmes programáveis por dia e capacidade de bloqueio através de chave, [14]. O dispositivo funciona alimentado através de bateria. (Figura 2.1a)
- **MedReady 16XX** – Este equipamento possui duas versões. A mais simples é muito semelhante à anterior. A mais elaborada possui um *modem* que permite enviar um relatório diário sobre o cumprimento das tomas através da internet. A sua alimentação principal é a rede elétrica; no entanto, possui uma bateria de reserva com 48 horas de autonomia. O preço destes equipamentos varia desde os 149 USD e 307 USD consoante as características escolhidas, [15]. (Figura 2.1b)
- **E-pill MedSmart MD2** – Este modelo destaca-se por possuir um *display* externo, que contém mais informações disponíveis do que os seus concorrentes. Este modelo tem duas variantes: a normal e a *plus*. A única diferença entre ambas é a capacidade de monitorização exterior. O sistema permite o envio dos relatórios das tomas, a programação remota do dispositivo e o envio de alertas sobre a necessidade de recarregamento. O preço é de 489,95 USD para a unidade sem monitorização e de 789,95 USD para o dispositivo com monitorização, [16]. (Figura 2.1c)
- **E-pill Med-O-Wheel SMART** – Este dispositivo destaca-se pela sua portabilidade, não possui monitorização e suporta 14 ou 28 tomas. Este dispositivo pode ter ou não chave de bloqueio e possibilidade de monitorização remota, [17]. Os seus preços são de 389,95 USD para a versão simples e de 489,95 USD para a versão com chave, [16]. (Figura 2.1d)

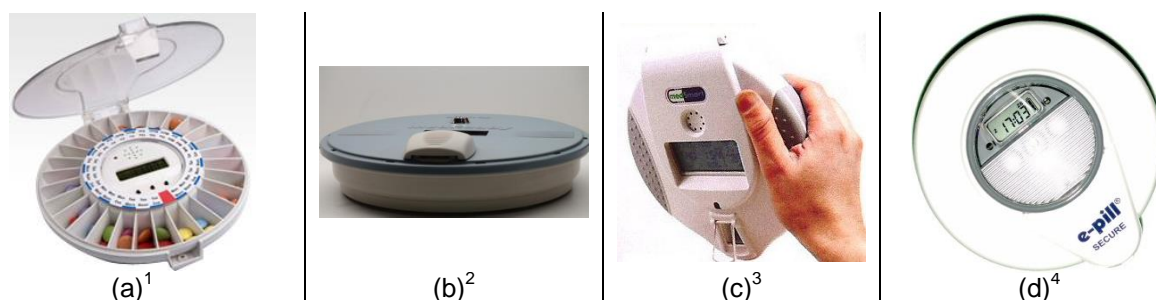


Figura 2.1 – Alguns dispensadores automáticos de medicação circulares;

Os dispositivos que serão referenciados de seguida têm formatos e particularidades que os distinguem dos anteriores:

- **E-pill MedSmart MD3** – Este dispositivo distingue-se pelo formato e pelo tabuleiro de tomas inviolável. Somente quando o tabuleiro da medicação é inserido na máquina e é bloqueado é que é possível retirar a tampa dele, mas continua a não ser possível o acesso aos medicamentos, a não ser quando o dispositivo o disponibilizar. Este equipamento não possui monitorização remota e o seu preço é de 844,95 USD, [16]. (Figura 2.2a)
- **Philips Medication Dispenser Resource Center** – Este dispositivo tem capacidade para 60 tomas; cada toma é guardada individualmente dentro de um pequeno copo de plástico com tampa; permite armazenar os medicamentos não tomados num reservatório à parte; permite avisar o utilizador para a toma de medicamentos que não sejam comprimidos ou cápsulas, (por exemplo, xaropes). Estes medicamentos não são armazenados no equipamento, mas o dispositivo pode avisar o utilizador que os deve tomar através da mensagem no seu ecrã e/ou através de voz, [18]. O preço de venda é de 844,95 USD; também tem disponível uma opção de aluguer, mas não foi possível identificar o respetivo custo, [16]. (Figura 2.2b)
- **MedMinder** – A empresa apresenta dois dispensadores: o *Maya* e o *Jon*. Os comprimidos são organizados numa matriz de 4x7 compartimentos, sendo os comprimidos colocados em pequenos copos. Não possui ecrã nem botões – toda a sua parameterização é realizada através de uma página *web*. O dispositivo, possui um *modem* GSM para enviar os registos das anomalias e receber a parametrização. A diferença entre os dois modelos é o facto de o *Jon* ter todos os compartimentos bloqueados, impossibilitando o acesso indevido às tomas, o que não acontece no *Maya*. A supervisão pode ser partilhada entre um prestador profissional de cuidados e, por exemplo, familiares. Este dispositivo só está disponível na modalidade de aluguer sendo o preço mensal de 39,99 USD e de 59,99 USD por mês, respetivamente para os modelos *Maya* e *Jon*. A empresa também disponibiliza um serviço de entrega de tabuleiros já carregados, simplificando a tarefa do utilizador, [19]. (Figura 2.2c)
- **Medimi** – Este dispositivo é o mais pequeno de todos os que já foram aqui referenciados. Foi concebido para ser portátil e cabe na palma da mão humana. Possui capacidade para 7 ou 15 tomas. Como meio de aviso o dispositivo possui alarme sonoro, luminoso e por vibração. Possui um módulo que possibilita comunicações via GSM e Bluetooth [20]. O seu preço de venda fica pelos 289,95 USD, [21]. (Figura 2.2d)

¹ Fonte: <http://ecx.images-amazon.com/images/I/41tYWxHmGrL.jpg>

² Fonte: <http://www.medreadyinc.net/cart2/images/medready-side.jpg>

³ Fonte: http://ep.yimg.com/ca//epill_2267_3865070

⁴ Fonte: http://ep.yimg.com/ca//epill_2268_4748475



Figura 2.2 – Alguns dispensadores automáticos de medicação não circulares;

Face ao exposto, podem elencar-se as seguintes funcionalidades mais relevantes destes equipamentos:

- Tabuleiro de tomas amovíveis;
- Monitorização remota com avisos (SMS, chamada ou correio eletrónico);
- Tabuleiro de tomas inviolável (depois de carregamento e antes de ser inserido no equipamento);
- Programação local através de botões e *displays*;
- Programação remota;
- Restrição no acesso ao medicamento com recurso a uma fechadura;
- Bateria de *backup*;
- Alarme sonoro, luminoso e por voz;
- Instruções de como efetuar as tomas no display e/ou por voz;
- Possibilidade de efetuar tomas de medicamentos que não sejam comprimidos ou cápsulas (xarope, pó, etc);
- Transportável, portátil;
- O número de compartimentos mais comum é de 28.

⁵ Fonte: http://ep.yimg.com/ca/l/epill_2272_45385936

⁶ Fonte: http://ep.yimg.com/ca/l/epill_2271_13550997

⁷ Fonte: <http://www.medminder.com/wordpress/wp-content/uploads/2013/03/jon.jpg>

⁸ Fonte: http://ep.yimg.com/ca/l/epill_2268_3553838

Capítulo 3 - Arquitetura do sistema

3.1 - Enquadramento

Na elaboração de projetos de engenharia complexos torna-se necessário implementar procedimentos e tarefas que ajudem a prevenir erros e que auxiliem a um planeamento sóbrio. Tendo isso em mente torna-se necessário elaborar uma estruturação rigorosa do projeto através da definição, descrição e concretização das partes integrantes do sistema. Torna-se assim necessário a criação de uma arquitetura concisa que torne a solução mais adequada para o problema em estudo.

3.2 - Arquitetura do sistema

Antes de iniciar a descrição da arquitetura do sistema desenvolvido há que indicar as linhas orientadoras que foram definidas depois de analisar os equipamentos disponíveis no mercado e depois da discussão com todos os intervenientes envolvidos no projeto. Os requisitos definidos para o sistema a desenvolver foram os seguintes:

- Sistema para uso doméstico, unipessoal e transportável;
- Capacidade para armazenar e dispensar entre 28 e 35 tomas;
- Possuir diversas formas de avisar o utilizador – por exemplo, visual e acústico;
- Permitir a monitorização e parametrização remota, através de, por exemplo, uma página *web*, que deverá permitir:
 - A definição dos horários de dispensa dos medicamentos;
 - A possibilidade da receção e visualização do cumprimento das tomas;
 - A parametrização de todas as variáveis do “sistema”.
- Ter como fonte de alimentação principal a rede elétrica; caso esta falhe também ter disponível uma bateria auxiliar que mantém a equipamento a funcionar durante, pelo menos, 48h.
- Possuir vários meios de comunicação que podem ser adicionados antes do comissionamento do equipamento. Os que poderão ser utilizados são:
 - GSM/GPSR/UTMS/LTE;
 - Wi-Fi (IEEE 802.11);
 - Ethernet (IEEE 802.3);
 - Bluetooth (IEEE 802.15).

Com base nos requisitos precedentes, foi desenhado um diagrama geral do sistema dispensador de medicamentos que pode ser visto na figura 3.1. Esta figura representa, de uma forma simplificada, todos os módulos do sistema e a forma como interagem entre si.

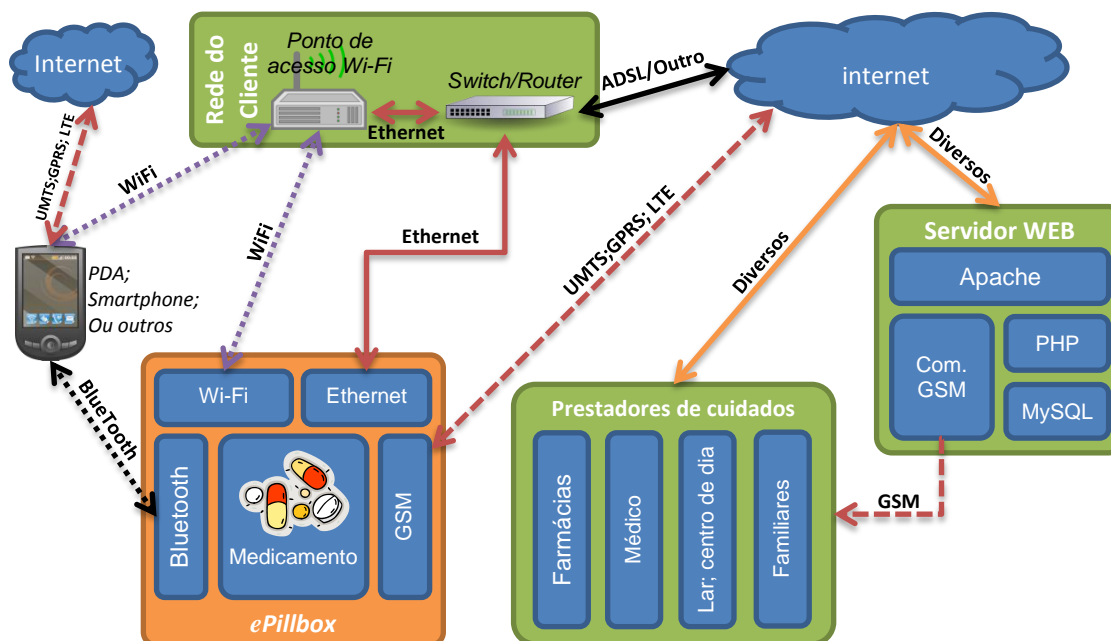


Figura 3.1 – Diagrama de blocos do sistema *ePillbox*.

Dado o carácter modular do dispositivo, antes de realizar o comissionamento do equipamento será necessário dotar o mesmo de, pelo menos, um meio de comunicação, cuja escolha será feita tendo em consideração as características e requisitos do cliente/paciente. Passada esta fase, o dispositivo está pronto a funcionar; após a fase de registar o dispositivo na aplicação *web*, e fazer a respetiva parametrização relativamente às tomas, definir os alarmes, entre outros.

O próximo passo consiste em carregar o dispositivo com os medicamentos nos locais certos, fechar o equipamento e ativá-lo. Então, o mesmo tentará comunicar com o servidor remoto para obter as respetivas parametrizações e horários das tomas. Como se pode observar pela figura 3.1 existem vários meios de aceder ao servidor remoto; dependendo dos módulos de comunicações a instalar, e dos meios a utilizar, a parametrização desses mesmos módulos será diferente. No caso da comunicação a utilizar ser:

- **GSM** – Nesta situação o cliente só precisará de colocar o equipamento onde tenha cobertura GSM para que o sistema funcione. Quando oportuno, o dispositivo ativará o módulo e ligar-se-á ao servidor remoto para trocar informação. Esta solução é muito prática já que não implica nenhuma configuração nem a aquisição de mais nenhum equipamento, por parte do paciente. No entanto, esta solução acarreta custos com as comunicações.
- **ETHERNET** – Esta é uma situação em que é necessário que o dispositivo esteja ligado por um cabo à rede. Também se torna obrigatório que o paciente possua uma rede *Ethernet* com ligação à Internet bem configurada. Nesta situação o equipamento, para comunicar com o servidor remoto, usa a rede do paciente. Este caso é o mais barato já que não acarreta custos adicionais, considerando que o paciente já tem infraestrutura de *Ethernet* a funcionar em casa. Tem outra vantagem que é o facto de não emitir radiação eletromagnética, por utilizar como meio de comunicação condutores elétricos.
- **WI-FI** – Este meio, em termos de infraestrutura, é muito semelhante ao anterior. A diferença está em que a comunicação é feita sem fios e que, por essa razão, é necessário um equipamento com o nome de ponto de acesso (A.P.). Tem a desvantagem, relativamente à anterior, o facto de ser necessário mais alguma configuração e de emitir radiações eletromagnéticas.
- **BLUETOOTH** – Nesta situação o equipamento comunicará por *Bluetooth* para um *smartphone* com uma aplicação desenvolvida para o efeito. Nesta aplicação será possível

a configuração do equipamento e também servirá de ponte para o servidor remoto. Utilizando a capacidade de comunicação instalada (GSM e Wi-Fi) no *smartphone* será possível o envio da informação para o servidor remoto.

Depois do problema da comunicação estar resolvido e do dispositivo estar corretamente configurado, será tempo de iniciar o ciclo das tomas. O equipamento permanecerá em repouso até que chegue à hora marcada para disponibilizar a toma. Chegado a esse ponto, o dispositivo, avisará o utilizador de diversas formas, mediante a configuração feita anteriormente. À medida que o tempo vai passando os alarmes vão-se intensificando. Caso o alarme seja bem-sucedido, o paciente chegará junto do aparelho, pressionará um botão e os medicamentos ser-lhe-ão disponibilizados. Esta é a situação normal e termina assim a rotina de entrega de medicamentos. De seguida o dispositivo ficará novamente em repouso a aguardar a próxima toma.

Caso o paciente não pressione o botão de dispensa de medicamento dentro do prazo indicado ser-lhe-á impedido o acesso e será gerado um alarme que será de imediato reportado ao servidor remoto. Caso a comunicação nesse momento não seja possível, o dispositivo irá tentar continuamente até conseguir entregar a informação ao servidor.

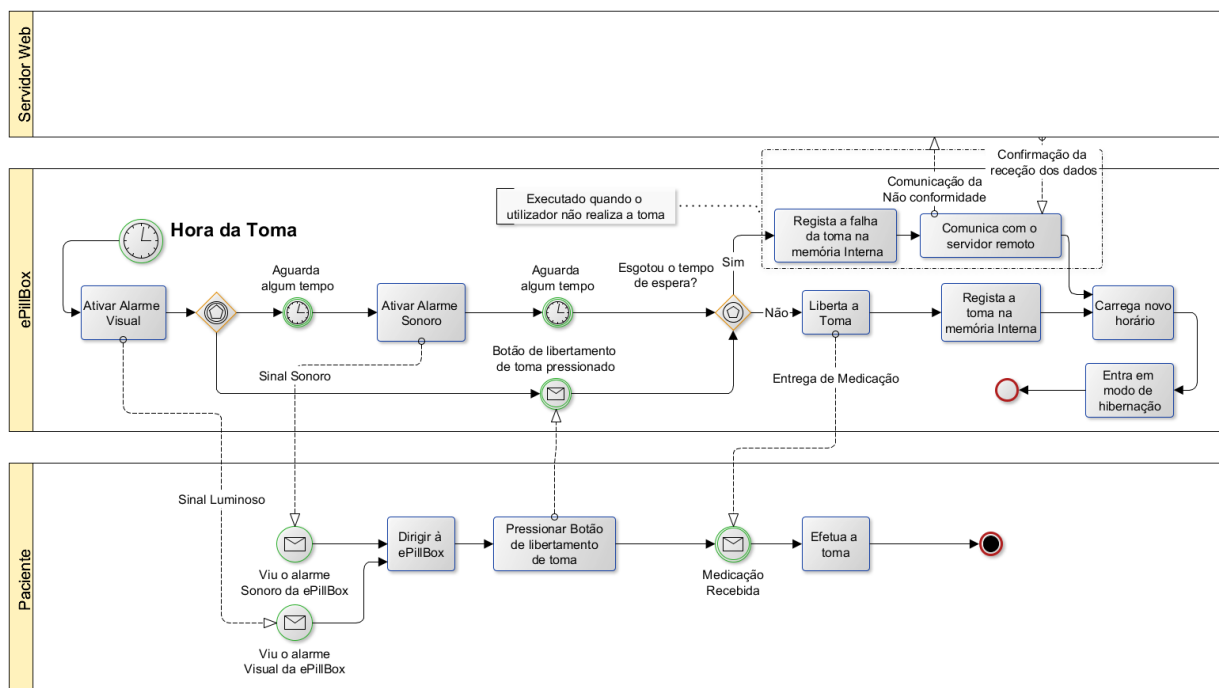


Figura 3.2 – Diagrama funcional da interação paciente, *ePillBox* e servidor remoto.

Caso o doente tome toda a medicação atempadamente será feito um envio, pelo menos uma vez por dia, para que essa informação fique disponível no servidor e também para se poder, mais facilmente, detetar situações anómalas com o dispositivo. A figura 3.2 apresenta um diagrama que ajuda a perceber melhor a interação paciente – *ePillBox* – servidor remoto;

O dispositivo será equipado com um botão de “pânico”, o qual, ao ser pressionado, fará com que seja feito um pedido para o servidor remoto e feito um contacto com o paciente para verificar a situação.

3.3 - Módulos do dispositivo *ePillBox*

Na figura 3.3 encontra-se um diagrama de blocos interno do dispositivo *ePillBox*. A nível da vertente eletrónica, o dispositivo é constituído pelos seguintes blocos: alimentação; microcontrolador; módulo de comunicações; interface com o utilizador; drivers dos atuadores mecânicos; atuador da abertura da toma; e por fim, sistema mecânico.



Figura 3.3 – Diagrama de blocos da *ePillbox*.

3.3.1 - Alimentação do sistema

Este módulo tem como função assegurar os valores de tensão para os restantes módulos do equipamento fazendo a conversão da tensão de 5V vinda da ficha *micro-USB* ou das baterias para 3,3V, que é a tensão utilizada para alimentar os restantes módulos.

A alimentação principal virá da rede elétrica que, através de um adaptador AC/DC, transforma os 230V AC para uma tensão DC adequada às necessidades do dispositivo. Em caso de falha da rede existe uma bateria que possibilita o funcionamento do dispositivo durante determinado intervalo de tempo. Outro dos componentes deste módulo é um carregador de baterias.

O tipo de baterias a utilizar poderá utilizar tecnologia que integre os seguintes elementos químicos: Ni-MH; Li-On; etc. Nesta fase do projeto poderá ser importante ter vários tipos de baterias para testar qual será a mais vantajosa para que o sistema cubra cabalmente as suas funções – 48 horas de autonomia, totalmente funcional.

3.3.2 - Unidade de processamento

Este bloco é responsável pela gestão de todo o dispositivo, possuindo os recursos e as funcionalidades necessárias para que o dispositivo opere nas condições especificadas. Os requisitos são os seguintes:

- Ter memória suficiente para guardar todos os dados necessários;
- Ter capacidade de processamento necessário para lidar com as diversas situações em tempo real;
- Possuir unidade de relógio de tempo real (RTC);
- Ter várias portas série (UART) para poder fazer *interface* com os diversos módulos de comunicação;
- Ter outros barramentos de comunicação série, tais como, por exemplo: I²C, SPI, SDIO, etc. para permitir a *interface* com os diversos periféricos incluídos no sistema;
- Possuir um ou vários modos de baixo consumo para minimizar o consumo total de todo o sistema;
- Possuir um conversor analógico-digital (ADC), para ler grandezas analógicas;
- Possuir uma unidade de modulação de largura de pulso (PWM) para controlar os atuadores de uma forma progressiva;

- Possuir portas digitais de uso geral (GPIO) em número suficiente para fazer a *interface* com os restantes módulos do sistema.

O microcontrolador escolhido foi um processador de 32 bits, de arquitetura RISC (*Reduced Instruction Set Computing*), o STM32F407VGT6 [22], da *STMicroelectronics*, com núcleo *ARM Cortex M4*. A tabela 3.1 apresenta as características mais importantes para o dispositivo que o microcontrolador possui.

Tabela 3.1– Resumo das características do STM32F407. [22, p. 1]

Especificações do microcontrolador STM32F407		
Tensão de alimentação		1,8 V até 3,6 V DC
Velocidade de relógio		Até 168 MHz
Memória interna	Flash	1 MB
	SRAM	192 kB
	Bakup	4 kB
Interfaces de comunicação		4 USART/, 2 UART, IrDA, I ² C, SPI, SDIO, I ² S, USB
Conversores de sinais		3 ADC, 2 DAC (12bits)
Outras		RTC, CRC, ID único, FPU

Algumas das razões que justificam a escolha deste microcontrolador estão expostas na tabela 3.1, na qual se constata corresponder às funcionalidades pretendidas. Pode dividir-se este processador em duas partes: o núcleo, e a integração do mesmo.

Relativamente ao núcleo do microcontrolador, este é um ARM Cortex M4. Ora, sendo a ARM (*Advanced RISC Machines*) uma empresa que desenvolve núcleos de processadores e, posteriormente licencia a sua tecnologia às empresas que produzem os chips, isto possibilita a escolha do fabricante que possui a implementação que for mais conveniente. Com a implementação feita pela *Cortex Microcontroller Software Interface Standard* (CMSIS) torna-se mais simples a portabilidade entre os fornecedores de chips [23, pp. 1-10]. A escolha do fabricante foi motivada essencialmente por dois factores:

1. Por este possuir uma grande variedade de MCU à escolha, todos com núcleo ARM Cortex-M, em que são quase 100% compatíveis pino a pino, sendo somente necessário realizar algumas alterações nos pinos de alimentação;
2. Por este disponibilizar um kit de desenvolvimento a um custo baixo, designado por *STM32F04 Discovery KIT* – menos de 12 Euros. [24]

O processador escolhido possui diversos periféricos internos que não serão utilizados, pelo que se poderia utilizar um MCU mais simples e, por consequência, com um custo inferior. No entanto, dado que se trata de um protótipo que servirá como base de testes, torna-se importante que seja mais poderoso do que o necessário, já que permitirá testar funcionalidades que precisarão de mais poder de processamento, mais memória, etc... – por exemplo decodificador de voz para avisos sonoros personalizados – MP3.

Tendo sempre em vista que se pretende também construir um produto comercial, onde o custo é sempre um fator importante, o desenho da PCI foi concebido para suportar vários MCUs. Assim, aquando da soldadura, através de *shunts* e da colocação ou não de alguns componentes, será possível a utilização de diversos processadores com funcionalidades e custos diferentes.

3.3.3 - Módulos de comunicação

Este módulo tem como função permitir que a *ePillbox* comunique com um sistema remoto onde será feita a gestão da informação. No diagrama de blocos (figura 3.1) pode ver-se que existe um conjunto de tecnologias diferentes de comunicação. Dado que o sistema é modular, o que se pretende é que o dispositivo tenha disponível pelo menos um meio de comunicação. As tecnologias GSM (*Glogal*

System for Mobile Communications), *Wi-Fi* (IEEE 802.11) e *Bluetooth* (IEEE 802.15) são tecnologias sem fios, sendo as duas últimas sem custos de operação ou licenciamento. A tecnologia *Ethernet* (IEEE 802.3) é a única, das apresentadas, que utiliza fios entre os equipamentos.

3.3.3.1 - GSM

A tecnologia GSM, apesar de ter custos, tem vantagens relativamente às outras apresentadas, tais como:

- Não necessita configurações por parte do utilizador;
- Não utiliza fios;
- Não necessita que o utilizador possua nenhuma infraestrutura para além da alimentação elétrica.

Não havendo necessidade do utilizador fazer configurações esta tecnologia pode tornar-se a melhor alternativa, apesar de ter custos associados.

3.3.3.2 - Wi-Fi /Ethernet

Para que a tecnologia *Wi-Fi* e *Ethernet* possam funcionar é necessário que o paciente possua uma rede *Ethernet* com acesso à *Internet* corretamente configurada. Mesmo assim será necessário realizar alguma configuração do equipamento. Dado que o público-alvo corresponde a pessoas idosas e/ou com algum tipo de limitação física ou psicológica, poderá inviabilizar este tipo de comunicação. No entanto, estes problemas poderão ser contornados se a instalação for executada por terceiros. Também poderá ser uma boa alternativa para lares de idosos, hospitais, etc...

3.3.3.3 - Bluetooth

Relativamente à comunicação via *Bluetooth*, esta serviria para fazer a *interface* com um *smartphone* com uma aplicação que serviria como *interface* para o equipamento. A comunicação com o servidor remoto seria feita através de *smartphone*. Nesta situação o *smartphone* serviria de “ponte” entre o equipamento e a internet onde reside o servidor remoto – meio de comunicação semi-definido na aplicação desenvolvida para *smartphone*.

3.3.4 - Interface com o utilizador

Este bloco tem como função avisar o utilizador de que uma nova toma está disponível. Este aviso pode ser luminoso e/ou sonoro. O aviso sonoro poderá ser um simples apito ou sons mais elaborados, como é o caso de uma melodia ou mesmo voz humana.

Também é necessário que haja possibilidade do paciente poder introduzir informação. Assim, o equipamento terá que possuir botões e sensores para cumprir essa tarefa. No mínimo, o equipamento terá que possuir:

- Botão de libertar a toma;
- Botão de pânico;
- Botão de *reset*/configuração do sistema;
- Sensor de deteção de gaveta;
- Sensor de abertura do compartimento dos medicamentos.

3.3.5 - Interface de potência com os atuadores

Dado que os atuadores, como é o caso dos motores elétricos, dos solenoides, etc., funcionam com gamas de tensões e correntes diferentes do microcontrolador, torna-se necessário introduzir um módulo no sistema que faça essa adaptação. Este módulo transforma os sinais de tensão e de corrente, vindos do processador, que são fracos, para uma tensão e corrente, mais elevados, compatíveis com os atuadores.

3.3.6 - Atuadores

Os atuadores são necessários para várias tarefas. Dependendo da sua função podem ser motores elétricos ou solenoides:

- No caso de tabuleiros circulares de tomas, com compartimentos radiais, é necessário, pelo menos, um motor para fazer rodar o mesmo e disponibilizar a toma. Também poderá ser necessário colocar um solenoide a fim de permitir, ou não, o acesso à toma;
- Caso o tabuleiro a utilizar seja retangular, com as tomas distribuídas por matriz, não será necessário nenhum motor. Terão que ser utilizados sinalizadores luminosos e poderão ser necessários solenoides para impedir que o utilizador aceda ao compartimento errado.

3.3.7 - Tabuleiro de tomas

Por fim, tem-se o tabuleiro das tomas. Este é o local onde é colocada a medicação dividida por tomas. O tabuleiro poderá ter dois formatos, um circular e outro retangular:

- O formato circular consiste num tabuleiro redondo compartimentado radialmente;
- No formato retangular os compartimentos estão distribuídos na forma matricial, tipicamente com quatro linhas e sete colunas. Neste formato cada compartimento tem uma tampa que protege os medicamentos e previne o acesso indevido. O número de compartimentos será preferencialmente múltiplo de sete, visto a semana ter sete dias, pelo que o presente equipamento terá entre 28 e 35 compartimentos.

3.4 - Sumário

Neste capítulo foi apresentada uma perspetiva geral do sistema que corresponde ao dispositivo implementado no âmbito do projeto de mestrado, as suas funcionalidades, bem como o funcionamento genérico do dispositivo.

Foi ainda apresentado um diagrama de blocos do sistema e uma descrição de cada um dos seus módulos, fazendo referência às características funcionais a que este tem de responder.

Capítulo 4 - Desenvolvimento do *hardware*

Este capítulo abrange todo o processo de criação do *hardware*. O processo inicia-se no dimensionamento e escolha dos componentes e termina no desenho da placa de circuito impresso, a qual será o meio físico que interligará os componentes de forma a permitir a disponibilização dos diversos serviços do equipamento.

4.1 - Enquadramento

O desenvolvimento de *hardware* consiste na criação de um esquema eletrónico com base nas especificações do sistema, apresentadas anteriormente. Um esquema eletrónico é um diagrama onde se recorre à utilização de um símbolo para representar o componente real e respetivas interligações.

Após a elaboração do esquema é necessário proceder a uma verificação de todas as ligações e opções feitas. Depois desse procedimento realizado com sucesso, o passo seguinte consiste no desenho da Placa de Circuito Impresso ou PCI. Aqui, os símbolos abstratos dos componentes são substituídos por representações bidimensionais, tendo as interligações que ser redesenhadas por forma a obter ligações curtas e com o menor número de vias possível. Após todo este processo, o passo seguinte consiste na geração de um conjunto de ficheiros que são entregues ao fabricante do *hardware* para produzir a respetiva PCI.

O pacote de *software* escolhido foi o “*KiCad EDA Software Suite*”. Trata-se de um *software* disponível sob a licença GNU-GPL V2, e que pode ser obtido sem custos, [25]. As principais razões para a sua escolha foram as seguintes: sem custos de aquisição; solução completa – captura do esquema e realização da PCI; grande disponibilidade de símbolos; e a possibilidade de pré-visualizar a PCI em 3D.

Dado que se pretende que o dispositivo seja muito flexível, permitindo testar várias configurações diferentes, o *hardware* desenvolvido foi dividido em vários módulos para que possam ser facilmente substituídos, caso seja necessário. Isto fez com que os custos fossem um pouco mais elevados do que se a placa fosse única – procurou-se um compromisso entre flexibilidade e custo.

4.2 - Módulo de Alimentação

A fonte de alimentação tem como função a adaptação dos níveis de tensão para os 3,3V com que funcionam todos os restantes componentes. O nível de tensão de entrada é de 5V vindos de um adaptador AC/DC externo. Na figura 4.1 pode ver-se um diagrama de blocos do módulo da fonte de alimentação.

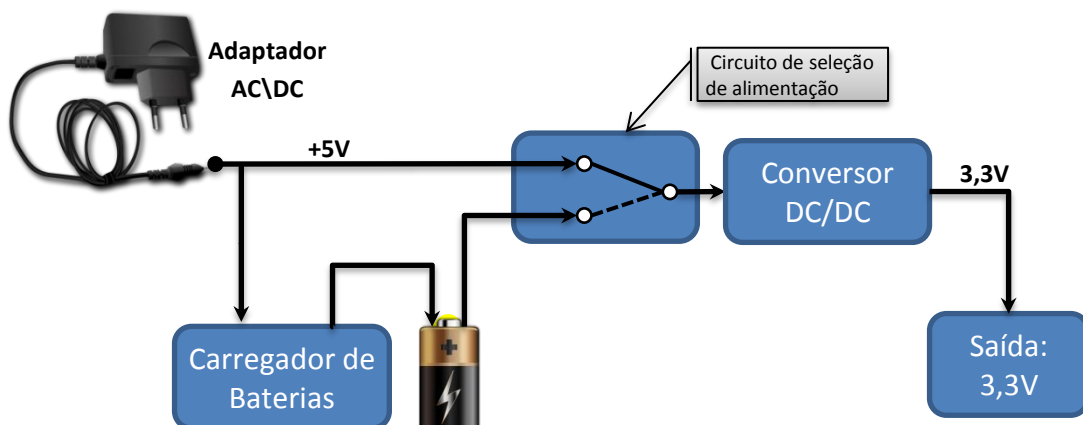


Figura 4.1 – Diagrama de blocos do módulo da fonte de alimentação.

4.2.1 - A alimentação externa

A entrada de alimentação é feita através de uma ficha micro-USB instalada no equipamento. Recorrendo a esta ficha é possível utilizar um adaptador *standard* AC/DC, utilizado, por exemplo, nos telemóveis. Isto faz com que seja fácil a substituição deste adaptador por outro no mercado.

4.2.2 - Conversor DC-DC

Para realizar a adaptação do valor da tensão de entrada (5V) para os 3,3V requeridos existem essencialmente três tecnologias:

- Reguladores lineares;
- Conversores comutados através de modulação por largura de pulso (PWM);
- Conversores comutados ressonantes de alta eficiência.

Cada uma das tecnologias possui pontos fracos e pontos fortes; não há uma tecnologia só com vantagens. Por consequência, importa ponderar adequadamente o que se pretende para se poder definir a fonte de alimentação mais adequada para o equipamento em causa. Na tabela seguinte comparam-se as principais características entre as diferentes tecnologias.

Tabela 4.1 – Comparação entre os 4 tipos de tecnologias de fontes de alimentação, adaptado de [26, p. 5]

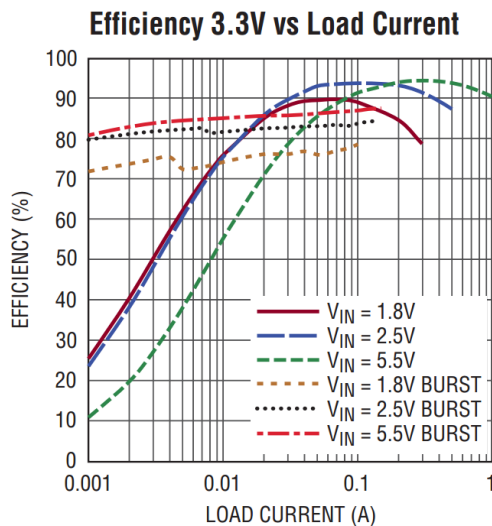
	Regulador linear	Conversores comutados por PWM	Conversores comutados ressonantes	Conversores comutados quase-ressonantes
Custo	Baixo	Alto	Alto	Alto
Massa	Alta	Baixa-média	Baixa-média	Baixa-média
Ruído RF	Inexistente	Alto	Médio	Médio
Eficiência	35-50%	70-85%	78-92%	78-92%
Saídas múltiplas	Não	Sim	Sim	Sim
Tempo de desenvolvimento	1 semana	8 Pessoas – mês ^a 5 Pessoas – mês ^b	10 Pessoas – mês ^a 8 Pessoas – mês ^b	10 Pessoas – mês ^a 8 Pessoas – mês ^b

^a Tendo um nível razoável de experiência facilitadas.

^b Com o uso deste livro [26]

Com base na tabela 4.1 optou-se pela utilização de um conversor comutado em vez do regulador linear dada a sua maior eficiência. Para que a fonte fosse mais compacta e simples de projetar optou-se pela utilização de um circuito integrado (CI) projetado especificamente para a função em causa.

No mercado existe uma variedade de CI com as mais diversas características, formas e preços. Quase todos os fabricantes de CI produzem conversores DC-DC. Tendo em consideração a eficiência, custo, tipo de pacote e disponibilidade nos fornecedores habituais (*Farnell*, *RS* e *Mouser*), a escolha recaiu sobre o CI LTC3536 da *Linear Technology*.



Lista de características do CI LTC3536:

- Gama de tensão de entrada / saída: 1,8V – 6V;
- Corrente de saída contínua: 1A;
- Eficiência máxima: 95%;
- Permite aumentar ou reduzir a tensão de saída comparativamente à entrada;
- Frequência de oscilação entre os 300kHz e os 2 MHz;
- 2 modos de funcionamento:
 - PWM – para melhorar o seu desempenho para uma corrente de saída baixa
 - Modo Burst – Melhora o desempenho em correntes de saída mais elevadas

Figura 4.2 – Características do CI LTC3536 [27, p. 4].

O LTC3536 permite manter o valor da tensão de 3,3V na saída independentemente do valor que está na sua entrada. Por outras palavras, o dispositivo permite fazer não só o abaixamento da tensão de 5 Volt, mas também, quando ligado a uma bateria com uma tensão inferior a 3,3V, permite manter a tensão de saída. Também permite modificar o modo de funcionamento de forma a manter a sua eficiência acima dos 80%, com altos ou baixos consumos.

A figura 4.3 ilustra o esquema do conversor DC-DC, o qual foi elaborado por um dos esquemas sugeridos em [27].

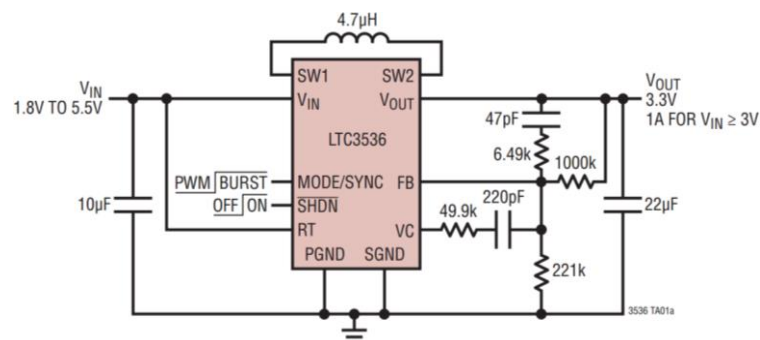


Figura 4.3 – Esquema do circuito do conversor DC-DC.

4.2.3 - Baterias

A bateria é um dispositivo que converte energia química, contida nos seus materiais ativos, em energia elétrica. Podem ser classificadas em primárias, quando não podem ser recarregadas, e em secundárias, no caso em que o processo químico pode ser revertido, designando-se a bateria por recarregável, [28, p. 1.3].

As baterias podem ainda ser divididas pelo tipo de elementos químicos utilizados no seu interior, tal como referido em [29]. Neste caso, a ênfase vai para as baterias recarregáveis, já que possuem um tempo de vida útil maior do que as primárias, uma vez que podem ser recarregadas várias vezes.

4.2.3.1 - A escolha do tipo de bateria

Sendo as baterias as principais fontes de energia para equipamentos eletrónicos, fazer a escolha correta do tipo de bateria a utilizar é um importante passo no processo de conceção de um equipamento.

Na escolha de uma bateria deve ter-se em atenção os seguintes fatores [30, p. 1]:

1. Densidade de energia (tamanho vs. peso);

2. Número de ciclos de carga/descarga (tempo de vida útil);
3. Capacidade (capacidade em manter o dispositivo funcional sem alimentação externa).

A tabela 4.2 apresenta uma análise comparativa entre os seguintes tipos de baterias: Alcalina (não recarregável); Chumbo selada (SLA); *Nickel-Cadmium* (Ni-Cd); *Nickel-Metal Hydride* (NiMH); Iões de lítio (Li-Ion).

Tabela 4.2 – Tabela com a análise comparativa de alguns tipos de baterias, adaptado de [30, p. 1]

Parâmetro	Química	Alcalina	SLA	NiCd	NiMH	Li-Ion
Densidade Energia/Peso ($W \cdot hr/kg$)		145	30-40	40-80	60-100	110-130
Densidade Energia/Volume ($W \cdot hr/dm^3$)		400	50-80	100-150	160-230	210-320
Voltagem de operação (V)		1,2	2,0	1,2	1,2	3,6
Voltagem em circuito aberto (V)		1,6	2,25	1,3	1,3	4,2
Voltagem quando descarregada (V)		0,9	1,75	0,9	0,9	2,8
Voltagem de carregamento (V)		N.A. ⁹	2,8	1,6	1,5	4,2
Auto descarga por mês (%)		0,3	2-8	15-20	20-25	6-10
Resistência Interna ($m\Omega$)		100-300	2,5-25	3,5-300	10-400	50-500
Ciclos carga/descarga		1	50-500	1500	800	1000
Taxa de descarga (mA. hr)		0,25C	<15C	<10C	<3C	<2C
Temp. de funcionamento (°C)		-20 – +55	-20 – +50	-20 – +60	0 – +60	-20 – +60
Custo inicial		Muito baixo	Baixo	Baixo	Médio	Elevado

Da análise comparativa apresentada na tabela 4.2 pode concluir-se o seguinte:

- Que se devem descartar as baterias alcalinas e as SLA, porque não são recarregáveis e por possuírem uma densidade de carga muito baixa, respetivamente.
- Que as baterias de NiCd são as que suportam mais ciclos de carga, são de baixo custo e possuem um algoritmo de carregamento muito simples; como desvantagens possuem efeito de memória, utilizam materiais tóxicos (*Cadmium*) e possuem uma baixa densidade de carga, quando comparada com as de NiMH e as de Iões de lítio.
- As baterias de NiMH podem ser consideradas um melhoramento das de NiCd; partilham as vantagens e desvantagem das de NiCd, mas com diferenças na capacidade, no número de ciclos e no algoritmo de carga.
- As baterias de Iões de lítio (Li-Ion) possuem o melhor desempenho nesta análise comparativa, com a vantagem de não possuírem efeito de memória; as desvantagens são o seu custo inicial mais elevado uma vida útil inferior em relação às anteriores. Também necessitam de circuitos de proteção contra sobretensões (carga) e subtensões (descarga). Ainda dentro das baterias de lítio existem as baterias de polímeros de lítio que são consideradas uma evolução das anteriores. Possuem formatos variados e espessuras pequenas, [30, p. 2].

⁹ Não Aplicável

Após esta análise, os tipos de bateria escolhidos para utilizar no equipamento foram as de *NiMH* e as de íões/polímeros de lítio; foram escolhidas as duas para que fosse possível testar ambas e, posteriormente adotar a mais adequada.

4.2.3.2 - Algoritmos de carga

O algoritmo escolhido para realizar a carga da bateria tem impacto na sua vida útil, fiabilidade e segurança. As baterias referidas no ponto anterior possuem algoritmos de carregamento diferentes, como se pode ver na figura 4.4.

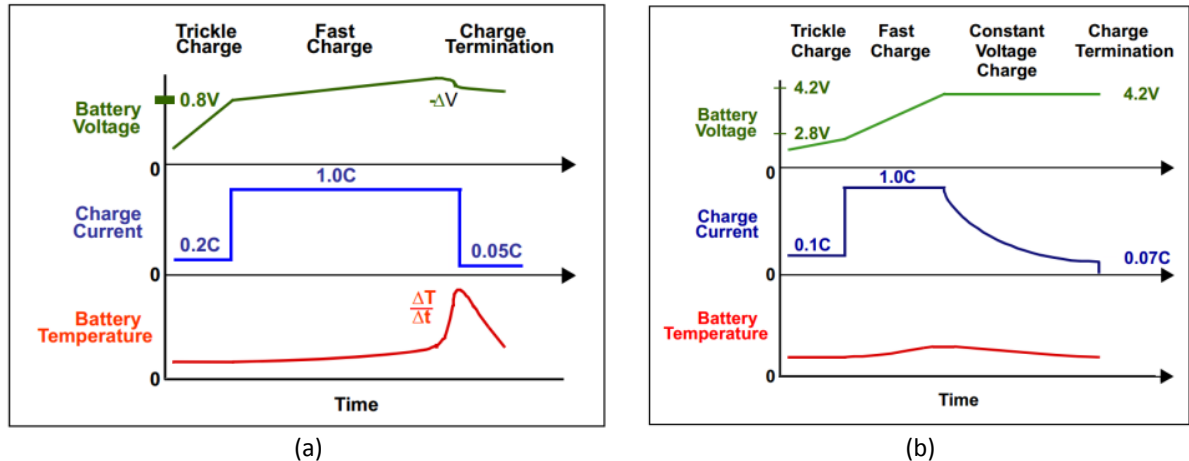


Figura 4.4 – Algoritmo de carga das baterias de *NiMH* (a) e *Li-Ion*(b), retirado de [30, pp. 3-4]

O algoritmo de carga mais simples para uma bateria de *NiMH* é aplicar-lhe uma corrente equivalente a 10% ou 20% da sua capacidade durante 10 horas. No entanto, este método de carga não é muito eficiente por demorar muito tempo e poder sobrecarregar as baterias. O método mais eficiente é o da figura 4.4 (a) que apresenta um algoritmo de três fases, e que garante uma carga rápida e segura deste tipo de pilha, [30, p. 3].

O algoritmo carga para as baterias de *Li-Ion* é mais complexo e baseia-se no método de “corrente constante ou controlada; voltagem constante” que pode ser visto na figura 4.4 (b).

Para ambos os tipos de baterias existem no mercado soluções de carregamento, como se pode observar nas referências [31-34].

4.2.4 - Circuito Carregador de Baterias

Neste caso, como se pretende utilizar dois tipos de tecnologias, optou-se pela utilização de uma solução integrada que permite carregar os dois tipos de baterias, baseada no circuito bq2000 da *Texas Instruments*, [33].

Nesta aplicação, devido à tensão mínima e máxima que o conversor DC-DC utiliza, ter-se-ia de utilizar uma bateria com:

- Uma célula para o caso de células de lítio ($V_{célua} = 4,2V$);
- Três células para o caso de células de *NiMH* ($V_{célua} = 1,2V \Rightarrow V_{3céluas} = 1,2 \times 3 = 3,6V$).

O circuito recomendado pelo fabricante pode ser visto na figura 4.5.

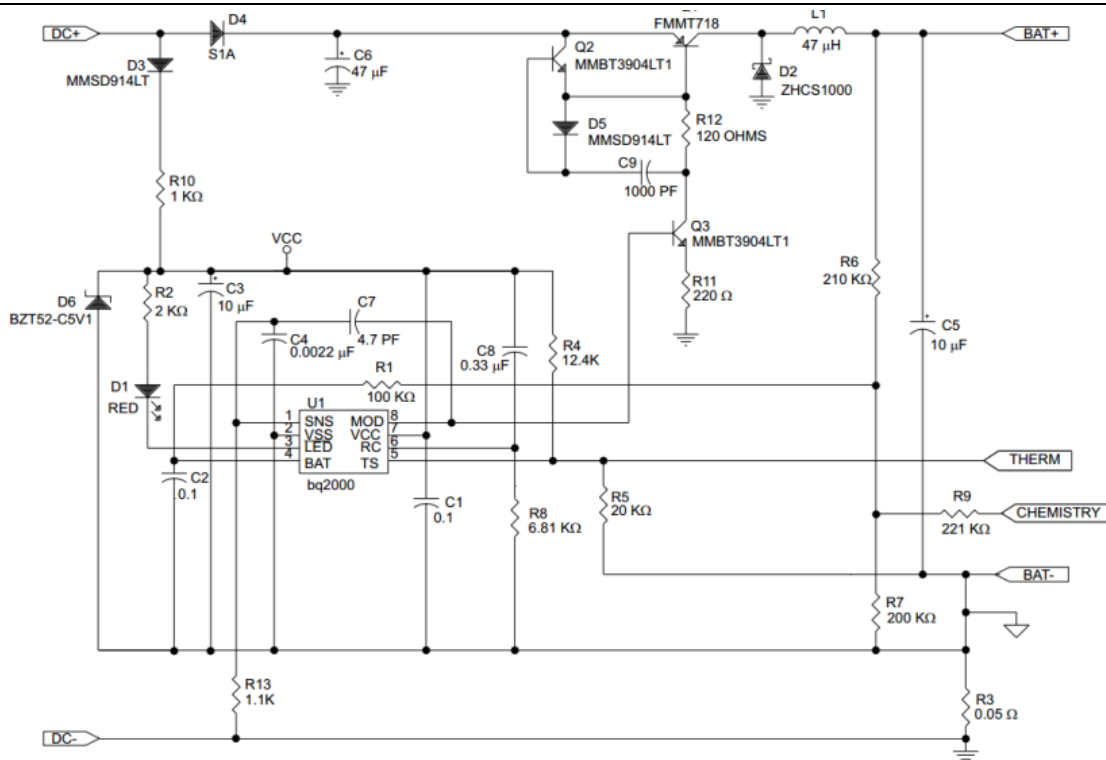


Figura 4.5 – Circuito de um carregador de 1A para uma célula de *Li-Ion* ou três células de *NiCd/NiMH*, retirado de [33, p. 10].

4.2.5 - Circuito de seleção de alimentação

Dado que o sistema de alimentação pode incluir duas potenciais fontes de tensão, torna-se necessário projetar um circuito que permita fazer a seleção entre elas; as duas entradas de alimentação são as seguintes:

- Alimentação principal, de 5V;
- Alimentação de reserva das baterias – esta pode oscilar desde os 4,2V das baterias de Lítio de uma célula, até cerca de 3V para um conjunto de três baterias de *NiMH* em série.

O circuito implementado foi o seguinte:

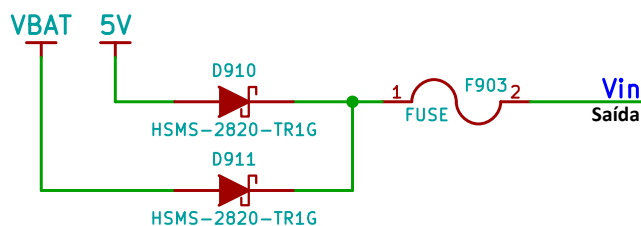


Figura 4.6 – Circuito do seletor de alimentação.

O circuito ilustrado na figura 4.6 tem o seguinte modo de funcionamento: Quando a alimentação de 5V está presente, esta polariza diretamente o diodo D910, e a corrente passa para a saída. Dado que a tensão da bateria é inferior a 5V, o diodo D911 fica inversamente polarizado, impedindo que a corrente da bateria seja consumida quando a alimentação de 5V está presente. Quando a tensão de 5V deixa de existir, o diodo D911 fica diretamente polarizado o que faz com que a corrente da bateria passe para a saída do circuito. Dado que D910 fica inversamente polarizado, não permite que a corrente da bateria passe para o lado dos 5V. Os diodos utilizados deverão ser diodos *Schottky*, com uma baixa de tensão de polarização, para evitar desperdícios de energia.

4.3 - Módulo do microcontrolador

Este é o módulo mais complexo do equipamento, já que é o centro de controlo de todo o dispositivo. O microcontrolador será o “cérebro” de todo o dispositivo, o que implica ter de se interligar com todos os outros dispositivos.

4.3.1 - O microcontrolador

O μC escolhido foi o STM32F407VGT6 que vem num pacote LQFP de 100 pinos, (figura 4.7).

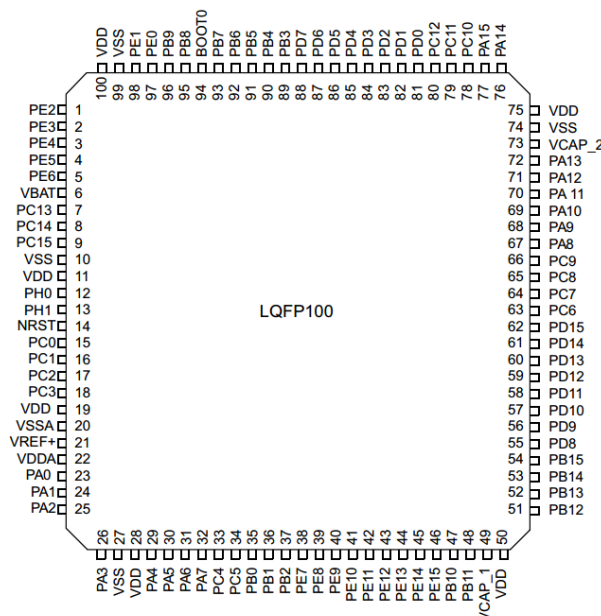


Figura 4.7 – Pinout do microcontrolador STM32F407VGT6 [22, p. 40]

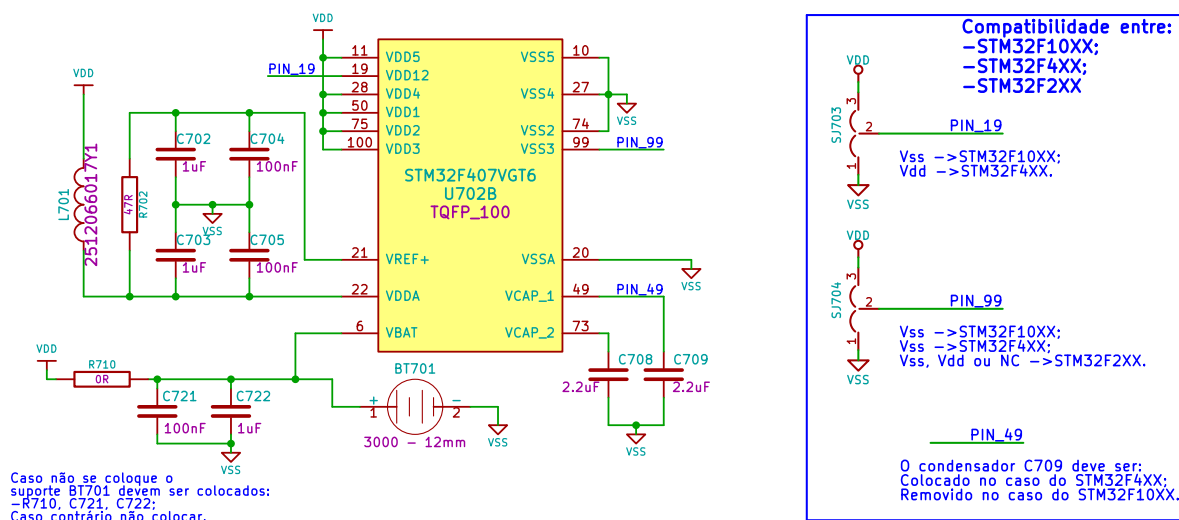
4.3.1.1 - Pinagem do microcontrolador

O encapsulamento do μC possui 100 pinos. Destes, alguns pinos têm uma função específica que não pode ser alterada, como é o caso dos pinos de alimentação (VDD, GND, Vbat, etc) ou o *Reset* (NRST), entre outros. Os restantes pinos, como é o caso dos GPIOs são multifunções, ou seja, podem ter várias funções, desde simples entradas/saídas, até funções especiais, como é o caso das *USART*, *SPI*, *I²C*, etc. Com o objetivo de otimizar os pinos e aproveitar ao máximo as suas potencialidades estes foram distribuídos pelos diversos periféricos mediante as características específicas de cada um. No anexo A está presente a lista dos pinos e respetivas funções atribuídas.

4.3.1.2 - Pinos de alimentação do μC

O μC possui diversos pinos de alimentação dispostos ao longo do CI; a figura 4.8 ilustra o esquema de ligação das alimentações do μC .

Tendo em consideração que este μC tem potencialidades acima das necessárias ao presente projeto, isto implica que seja mais dispendioso do que outras versões com menos capacidades, pelo que a PCI foi desenvolvida de forma a poder levar outros processadores da mesma família, na altura da montagem dos componentes. Como nem todos os μC da família STM32F são compatíveis pino a pino, principalmente nas alimentações, foram deixadas opções na PCI para possibilitar a troca de μC . Na figura 4.8, do lado direito, está exposta essa opção nos pinos 19, 49 e 99.

Figura 4.8 – Esquema de alimentação do μC .

Na conceção do esquema de ligações foram tidas em consideração as recomendações que constam no respetivo *datasheet* [22, p. 75] e no esquema da placa de desenvolvimento deste microcontrolador [35, p. 3].

Foi também adicionado um suporte para colocação de uma pequena pilha para alimentar a área de *backup* do μC – a área de *backup* compreende uma zona de memória RAM de 4K e o RTC, mantendo-se em funcionamento mesmo que a alimentação principal do μC não esteja disponível. A tensão a utilizar (V_{bkp_Bat}) deve situar-se entre 1,65V e 3,6V. Segundo [22, p. 90], o consumo máximo, mantendo ativos todos os periféricos que constituem a área de memória, é de $2\mu A$, considerando uma temperatura de funcionamento de $40^{\circ}C$. Os componentes R710, C721 e C722 só devem ser colocados caso não se pretenda colocar a pilha de alimentação da área de *backup*.

4.3.1.3 - Entradas Analógicas

As entradas analógicas têm como função a medição de grandezas analógicas. Ao contrário das grandezas digitais, que só possuem dois valores diferentes (ex. 0/1; Falso/Verdadeiro, etc...), as grandezas analógicas podem variar entre um número infinito de estados.

Tendo como objetivo realizar um melhor controlo do equipamento torna-se necessário medir o valor de algumas tensões no circuito; as grandezas a medir estão presentes na tabela 4.3, bem como o respetivo pino utilizado no μC .

Tabela 4.3 – lista de pinos analógicos utilizados.

Pino no μC	Nome pino	Nome Função	
31	ADC12_IN6	V_{USB}	Medição da tensão de alimentação do circuito;
26	ADC123_IN3	I_Ctrl_Tampas	Medição da corrente consumida pelo atuador da tampa quando ativado;
35	ADC12_IN8	V_{BAT}	Medição da tensão da bateria.

Circuito de Medição da tensão V_{USB} :

A tensão V_{USB} pode variar de 4,45V a 5,25V, segundo as especificações da norma USB 3.0 [36, p. 10]; contudo, o ADC não pode medir tensões superiores à tensão de referência utilizada, que neste caso é a mesma da alimentação (3,3V). Para contornar esta limitação, foi necessário a colocação de um circuito que consiga reduzir a tensão para a gama pretendida. O circuito mais simples que se pode utilizar é o divisor de tensão, o qual cumpre a função requerida. O objetivo principal desta entrada é a de permitir ao μC saber se há alimentação disponível na porta *micro-*

USB. Caso esta não esteja disponível, isso significa que o sistema está a ser alimentado pela bateria (figura 4.9).

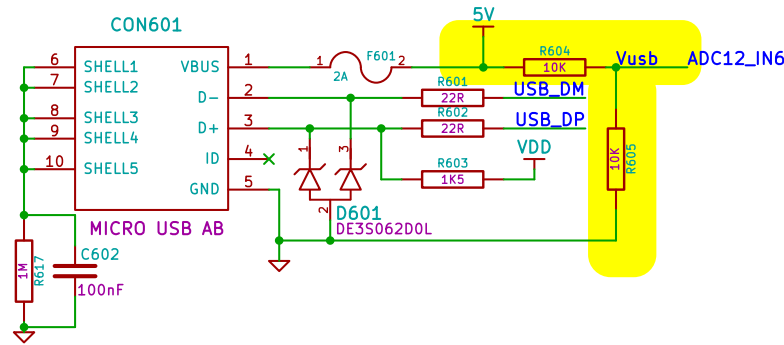


Figura 4.9- Esquema da ligação da porta *micro-USB* ao μC .

Dimensionamento do divisor de tensão: V_{USB}

Considerando que o pino do ADC consome uma corrente desprezável, então o circuito apresentado na figura 4.9 pode ser considerado um divisor de tensão.

A fórmula do divisor de tensão é:

$$V_{USB} = \frac{R_{605}}{R_{604} + R_{605}} \times V_{CC} \quad (4.1)$$

O valor da tensão máxima desejada tem que ser menor ou igual a 3,3V (tensão de referência do ADC), o qual não pode ser muito baixo para não se perder resolução. Uma boa redução poderá ser para metade, ou seja:

$$V_{USB_{MAX}} = \frac{1}{2} \times V_{CC} \Rightarrow \frac{1}{2} \times 5,25V = 2,625V \quad (4.2)$$

Considerando as equações (4.1) e (4.2) facilmente se compreende que:

$$\frac{R_{605}}{R_{604} + R_{605}} = \frac{1}{2} \quad (4.3)$$

Esta equação obtém-se fazendo com que $R_{604} = R_{605}$. Neste caso, foi escolhida uma resistência de $R_{604} = R_{605} = 10k\Omega$. Este valor é suficientemente elevado para não causar um consumo excessivo de corrente ($< 300\mu A$) e não ser demasiado grande para poder causar ruído ou mau funcionamento ao ADC.

Circuito de Medição da V_{BAT} :

A tensão V_{BAT} pode variar dentro dos valores V_{min} e V_{max} ; dependendo da tecnologia utilizada no fabrico das baterias, e os respetivos valores podem ser encontrados na tabela 4.4. Dado que a tensão continua a ser mais elevada do que a tensão de referência do ADC é necessário baixá-la para um valor que fique dentro da sua gama de conversão do ADC. O método utilizado foi o mesmo do da adaptação do sinal V_{USB} . O circuito com os respetivos valores está ilustrado figura 4.10.

Tipo de tecnologia	V_{min}	V_{max}
Bateria de Li-Ion (1 célula)	2,8V	4,2V
Bateria de NiMH (3 Células)	2,7V	3,9V

Tabela 4.4 – Valores mínimos e máximos de tensão das baterias segundo o tipo de tecnologia utilizada no seu fabrico.

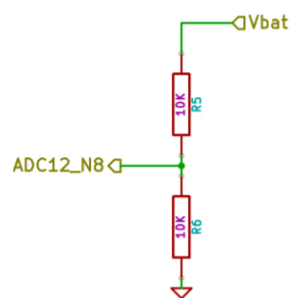


Figura 4.10 – Esquema de Medição da tensão da
bateria V_{BAT} .

Circuito de Medição da: $I_{ctrl\ Tampas}$

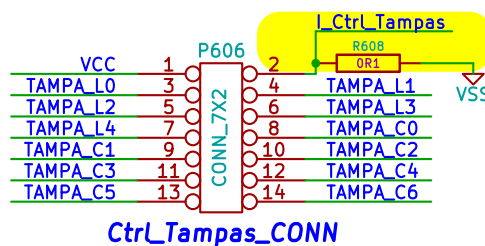


Figura 4.11 – Circuito de medição de corrente proveniente do módulo de controlo de Tampas

Atendendo a que o ADC não pode medir corrente, só consegue medir tensões, é necessário implementar um circuito que efetue a conversão da corrente para tensão. O sistema mais simples que se pode utilizar é uma resistência. Este componente, quando percorrido por uma corrente, produz uma queda de tensão entre os seus terminais. Através da lei de Ohm é possível obter uma relação entre a corrente e a queda de tensão.

$$U_R = R \times I \Leftrightarrow I = \frac{U_R}{R} \quad (4.4)$$

Considerando que a corrente máxima recomendada (I_{MAX}) que o atuador de abertura das tomas suporta é de $400mA$, a corrente máxima que passará na R_{608} é a mesma, logo,

$$U_{R_{MAX}} = R \times I \Rightarrow 0,1 \times 0,4 = 0,04 = 40mV \quad (4.5)$$

$$P_{R_{MAX}} = R \times I^2 \Rightarrow 0,1 \times 0,4^2 = 0,016 = 16mW \quad (4.6)$$

A resolução máxima do ADC é de 12bits e, considerando a tensão de referência como sendo 3,3V, tem-se uma tensão em degrau:

$$U_{\text{Step}} = \frac{V_{\text{REF}}}{2^{N_{\text{bits}}}} \Rightarrow \frac{3,3}{2^{12}} \cong 0,00081 = 0,81\text{mV} \quad (4.7)$$

$$V_{\text{ADC}_{\text{MAX}}} = \frac{U_{\text{R}_{\text{MAX}}}}{U_{\text{Step}}} \Rightarrow \frac{0,04}{0,00081} \cong 49 \quad (4.8)$$

Assim, o valor obtido pelo ADC varia de 0 a 49. Para aumentar a gama dinâmica do ADC poder-se-ia aumentar a tensão de entrada no ADC, o que pode ser obtido de duas formas:

- Aumentando a resistência em série com o circuito;
- Mantendo a resistência e colocando um circuito amplificador.

A primeira opção tem o inconveniente, se for muito elevada, de não permitir que passe a corrente recomendada pelo atuador de abertura das tomas fazendo com que este demore mais tempo a atuar. A segunda opção não tem esse inconveniente; no entanto, como se tem que adicionar mais componentes à placa, estes irão onerá-la.

4.3.1.4 - Circuito de Reset do μC

Para permitir o *reset* manual do μC foi instalado um botão para o efeito. O botão só estará disponível no interior do equipamento. A resistência R_{701} faz de *pull-up*, uma vez que o *reset* é ativo no nível lógico 0, e como o botão SW701 é normalmente aberto (NA), o pino NRTS fica com um nível lógico 1 quando este não está pressionado. O condensador C_{701} foi colocado para suprimir o ressalto (*bouncing*) do botão de *reset* quando este é libertado e criar um atraso quando o circuito é alimentado; o μC é mantido em modo *reset* durante mais uns instantes para dar tempo à alimentação de estabilizar. Na figura 4.12 pode-se ver o esquema do *reset* do μC .

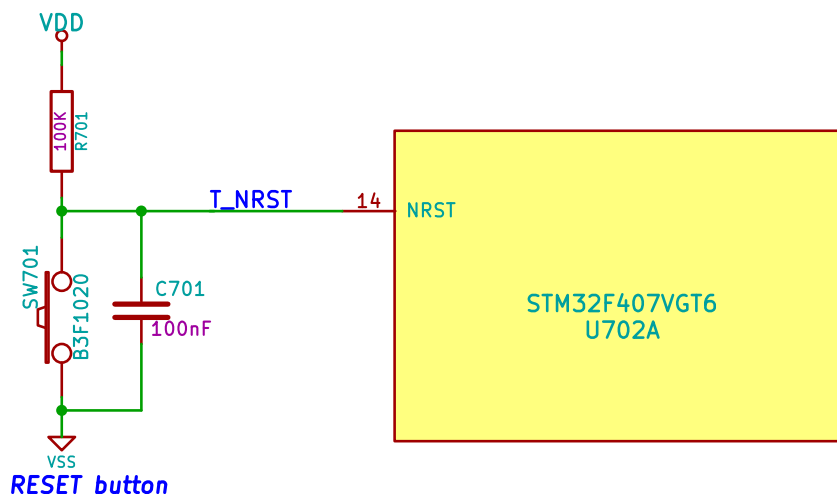


Figura 4.12 – Esquema de ligação do botão de reset ao μC

4.3.1.5 - Interface de programação SWD (Serial Wire Debug).

Para que seja possível a programação do μC foi incorporada uma porta compatível com o SDW. Com esta não só é possível a programação do dispositivo, mas também realizar o *debug* do *firmware* recorrendo a *software* e *hardware* especializado. A *interface* SWD permite funcionalidades semelhantes ao JTAG, mas utiliza um conjunto de pinos do μC mais reduzido.

Tabela 4.5 – Listagem dos pinos que contêm o interface SWD

Nome Pino SWD	Nº Pino	Tipo	Função	Nome do pino μC
VDD	1	-	Pino de Alimentação	---
SWCLK	2	E ¹⁰	Pino de Relógio	PA14
GND	3	-	Massa	---
SWDIO	4	E/S ¹¹	Pino de dados (E/S)	PA13
NRST	5	E	Reset do μC	NRST
TRACESWO	6	-	TRACESWO se a função de trace assíncrono estiver ativado	PB3

¹⁰ E: Entrada;

¹¹ S: Saída;



Figura 4.13 – Esquema da ficha SWD e interligação com o programador incluído na placa de desenvolvimento "STM32F4-Discovery";

A tabela 4.5 indica a funcionalidade de cada um dos pinos da *interface* ST-Link/SWD. A figura 4.13 ilustra o esquema de ligação do módulo do μC e um programador, que neste caso é a placa de desenvolvimento "STM32F4 – Discovery". Esta contém um programador que pode ser utilizado para programar o chip na placa de desenvolvimento ou um chip externo através do conector situado no lado esquerdo da placa. A secção da porta SWD ou o chip na placa são ligados nos 2 *jumpers* situados no lado direito da placa: Se estiverem colocados, a *interface* dará acesso ao microcontrolador da placa; se estiverem removidos, a *interface* dará acesso à porta SWD. Para mais informações relativamente à *interface* SWD e às funcionalidades de *debug* do μC consultar [37, pp. 1655-1686].

4.3.1.6 - Opção de Arranque de sistema

Este μC possui vários modos de arranque que estão presentes na tabela 4.6 [37, pp. 69,70]. Nesta aplicação os modos que se utilizam são os seguintes:

- Arranque pela memória principal;
- Arranque pela memória de sistema.

A seleção desses modos é feita através do *jumper* JP701. Quando este estiver desligado o μC arranca pela memória *flash* onde está armazenado o *firmware* do equipamento. Quando está ligado é selecionada a memória PROM que contém um *software* que permite a programação do μC através das *interfaces* de comunicação do mesmo [38, pp. 71-79]. Este modo permite a programação do dispositivo sem recurso a um programador dedicado. Através deste modo, recorrendo à ficha micro-USB instalada e que está disponível na parte de fora do equipamento, é possível atualizar o *firmware* recorrendo a um PC com *software* específico e a um cabo USB [39].

Tabela 4.6 – Modos de arranque

Pinos de seleção de modo de arranque		Modo de arranque	Descrição
Boot1	Boot0		
X	0	Memória Flash principal	A memória principal fica selecionada para memória de arranque.
0	1	Memória PROM de sistema	Memória PROM de sistema fica selecionada para memória de arranque.
1	1	SRAM Interna	SRAM Interna fica selecionada para memória de arranque.

Como seria de esperar, o modo de funcionamento normal será o arranque pela memória principal. Na figura 4.14 pode ver-se que, quando o *jumper* JP701 não estiver ligado, por defeito, o pino BOOT0 ficará com o valor lógico 0 e o μC arrancará pela memória *flash* principal; este *jumper* não vai estar disponível no exterior do equipamento. A sua utilização será somente necessária em último recurso, quando todos os outros métodos falharem. O que quer dizer que esta opção não estará acessível ao utilizador do equipamento, mas sim ao técnico que terá de fazer uma intervenção no equipamento.

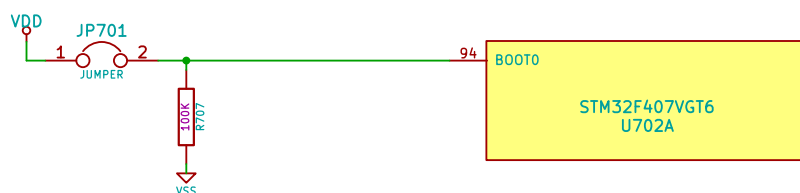


Figura 4.14 – Esquema de ligação do pino Boot0 μC

4.3.1.7 - Fontes de relógio externo

Todos os μC , por serem circuitos sequenciais, necessitam de uma fonte de relógio para poderem funcionar. Este microcontrolador permite receber sinais de relógio internos e externos. Os internos têm como vantagem não necessitar de nenhum componente externo e não ocupar pinos, ficando assim com mais portas disponíveis. Para além disso possuem uma desvantagem muito grande, que é o facto de terem uma precisão limitada e serem muito sensíveis a variações de temperatura e a variações de tensão. Adicionalmente, os osciladores externos à base de cristais de quartzo são muito mais precisos que os osciladores internos. Basicamente, as vantagens destes são as desvantagens dos internos e vice-versa.

Neste projeto decidiu-se utilizar osciladores à base de cristais de quartzo. Para a criação do relógio principal do sistema foi utilizado um cristal de 8MHz. Como neste caso se precisou de utilizar o RTCC (*Real Time Clock Calendar*), ou, em português, Calendário/Relógio de Tempo Real, que tem que ser o mais preciso possível e continuar a funcionar em modo de hibernação mantendo um consumo baixo, foi também incorporado um cristal de baixa frequência de 32,768kHz. Na figura 4.15 está presente o circuito de ligação dos cristais ao μC .

Nos documentos técnicos do μC , o oscilador principal é mais rápido, e tem por nome HSE (*High Speed External*) e o oscilador mais lento de LSE (*Low Speed External*) [37, pp. 149-156]

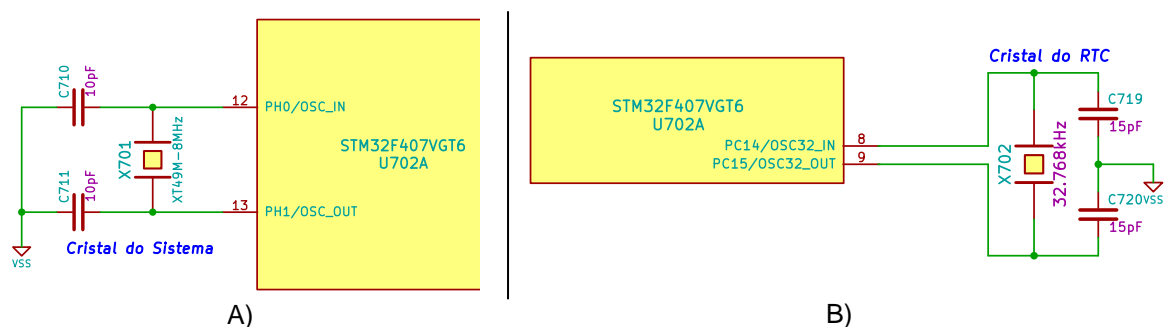


Figura 4.15 – Circuito dos osciladores externos (a:HSE; b:LSE)

4.3.2 - Chip de expansão de portas

A introdução deste *chip* não tem a ver com a falta de pontos de entrada/saída do μC , mas sim com duas limitações presentes no dispositivo quando este está no modo mais profundo de hibernação:

1. A primeira tem a ver com o facto do μC , quando no modo de hibernação mais profundo (no qual tem o consumo mais baixo), coloca todas as *GPIO* em modo de alta-impedância, o que impossibilita o μC de manter pinos ativados (quer com nível lógico alto ou baixo) – Isto impede a continuação do funcionamento de alguns dispositivos, como é o caso do carregador de baterias.
2. A segunda limitação prende-se com o facto de que, neste modo, as fontes de interrupção que acordam o μC são muito limitadas. A nível externo só pode ser acordado de duas maneiras:
 - Pelo pino de *NRST*;
 - Pelo pino *WKUP* (PA0) [37, p. 134].

Para suprir estas duas limitações foi colocado o Circuito Integrado (CI) MCP23008.

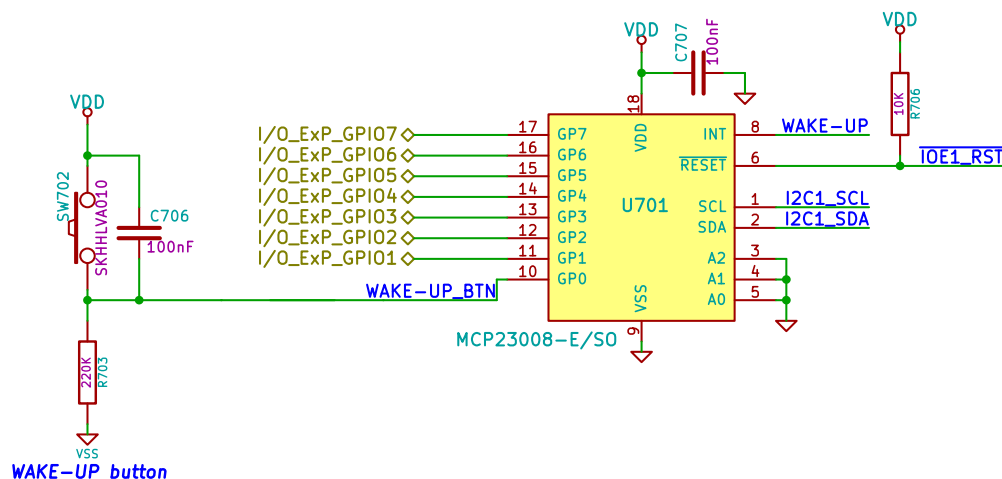


Figura 4.16 – Esquema do chip MCP23008

O MCP23008 possui oito pinos de entrada/saída com o objetivo geral de comunicar com o μC , utilizando o barramento de comunicação I^2C . Para além dos pinos utilizados para o barramento I^2C (SCL e SDA) também estão ligados ao processador os pinos números oito (INT) e seis (RESET) [40, p. 5]. O pino de reset, como o próprio nome indica, serve para efetuar um reset ao CI e o INT, tendo como função, depois de corretamente configurado, de enviar um sinal para o μC para o informar de que algumas das suas entradas modificaram o seu estado. Esta última funcionalidade é a mais importante e permite que o μC seja colocado no modo mais profundo de hibernação; quando uma das entradas mudarem de estado fará com que o μC acorde e tome conta da ocorrência.

Foi ainda colocada uma resistência de *pull-up* (R706), como se pode ver na figura 4.16, para evitar o reset do CI quando o μC entrar em modo de hibernação mais profundo. Uma vez que já foi dito anteriormente que quando este entrasse no modo de hibernação mais profundo todas os seus pinos de E/S ficariam em modo de alta-impedância, perdendo assim o valor lógico alto (3,3V).

Tabela 4.7 – Lista de funções das E/S do CI MCP23008 para e respetiva localização nos conetores

Nome pino	Pino do C.I.		Função	Conector	
	Nome	Nº		Nome	Pino
WakeUp_BTN	GP0	10	Botão de configuração	---	--
PWR_GPIO0	GP1	11	Controlo do modo de operação do conversor DC/DC	Power_CONN	10
PWR_GPIO1	GP2	12	Controlo do CI de carregamento da bateria	Power_CONN	11
PWR_GPIO2	GP3	13	Indicação da Carga terminada	Power_CONN	12
I/O_Exp_GPIO4	GP4	14	Sensor de porta de tomas aberta	I/O_CONN	8
I/O_Exp_GPIO5	GP5	15	Sensor de deteção da gaveta das tomas	I/O_CONN	7
I/O_Exp_GPIO6	GP6	16	Botão de Pânico	I/O_CONN	6
I/O_Exp_GPIO7	GP7	17	Botão de toma	I/O_CONN	5

A Tabela 4.7 ilustra a atribuição de funções aos diversos pinos de E/S do CI e também a localização desses mesmos pinos nos conectores do módulo do μC . Com exceção do botão de configuração todas as outras E/S estão implementadas fisicamente fora do módulo do μC .

Para terminar, o endereço do CI no barramento do I^2C é o 0x40, porque todos os pinos que estão no endereço (A0, A1, A2) foram colocados a zero [40, p. 8].

4.3.3 - Slot de cartões *MicroSD*

Este μC possui uma *interface* SDIO (*Secure Digital Input Output*) embutida permitindo assim a ligação direta a um cartão SD (*Secure Digital*). Para tal basta colocar um suporte de cartões, umas resistências de *pull-up* e um condensador de desacoplamento, obtendo-se um leitor de cartões *MicroSD*.

Um cartão de memória *MicroSD* é uma memória *Flash* amovível e substituível que necessita de um leitor/suporte que obedece ao seguinte padrão de cartões portáteis, tabela 4.8:

Tabela 4.8 – Padrão de cartões MicoSD.

Especificações	Cartão de memória <i>MicroSD</i>
Largura	11mm
Comprimento	15mm
Espessura	1mm
Volume	165mm ³
Peso	Aproximadamente 0.5g
Tensão	2.7 – 3.6V
Proteção contra gravação	Não
Proteção nos terminais	Não
Número de pinos	8 Pinos

Uma memória *Flash* é uma memória não volátil, ou seja, guarda o conteúdo mesmo sem fonte de alimentação, tipo EEPROM, que apresenta, nos dias de hoje, capacidades que variam de poucos *kBytes* até a alguns *GBytes*. Tem uma limitação importante que é a do número de leituras/escritas ser finito, o que significa que, após um uso extenso, alguns sectores podem ficar danificados e inutilizáveis.

A inclusão de um leitor de cartões permite expandir a memória do sistema. Uma das utilidades pode ser a inclusão de capacidade de leitura e reprodução de ficheiros de áudio com a finalidade de tornar os alarmes sonoros mais úteis e agradáveis.

Na tabela 4.9 está presente uma listagem dos pinos e funcionalidades da *interface* SDIO; na figura 4.17 está o esquema de ligações do mesmo.

Tabela 4.9 – Lista das Correspondência entre o interface SDIO e o μC .

SDIO	Função	Pino μC	
		Nome	Num
SDIO_D0	SD Serial Data 0	PC8	65
SDIO_D1	SD Serial Data 1 (memory cards) Interrupt Period (SDIO cards share pin via protocol)	PC9	66
nIRQ			
SDIO_D2	SD Serial Data 2	PC10	78
SDIO_D3	SD Serial Data 3	PC11	79
SDIO_CMD	Command, Response	PD2	83
SDIO_CK	Serial Clock	PC12	80
SDIO_CD1	SD card Detect	PA15	77

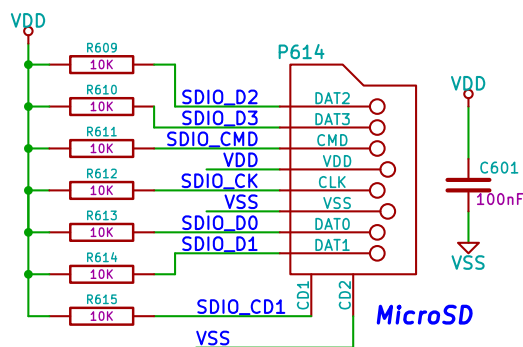


Figura 4.17 – Circuito de ligação do suporte do leitor de cartões micro-SD e o μ C.

4.3.4 - Porta *micro-USB*

Foi adicionada uma porta USB com dupla funcionalidade:

1. A primeira é de servir como entrada de alimentação para o equipamento;
2. A segunda serve para efetuar a atualização de *firmware* do dispositivo e, possivelmente proporcionar a realização de *debug*.

A alimentação poderá ser providenciada por dois dispositivos diferentes:

1. O primeiro será pelo PC, que poderá fornecer até 500mA;
2. O segundo poderá ser um adaptador AC/DC com ficha *Micro-USB*, sendo que, estes últimos poderão fornecer uma corrente superior a 1A.

Terá que ser implementado um algoritmo no μ C para verificar qual o equipamento que está a fornecer a alimentação. Caso seja o PC a fornecer energia, onde a corrente máxima fornecida é de 500mA, terão que ser tomados cuidados para não ligar determinados componentes do equipamento para não exceder a corrente máxima – só quando for ligado o adaptador AC/DC é que todo o dispositivo poderá funcionar sem limitações.

A deteção do adaptador AC/DC será feita tendo em consideração que este apresentará uma resistência inferior a 200 Ω entre os pinos D+ e D-, conforme o indicado em [41, p. 3].

Tabela 4.10 – Lista das Correspondência entre o interface USB e o μ C

MicroUSB	Função	Nome	Num
VBUS	Alimentação de 5V	-	-
D-	Pino de dados -	PA11	70
D+	Pino de dados +	PA12	71
ID	Permite distinguir entre uma ligação de <i>host</i> ou uma ligação de <i>slave</i> : <ul style="list-style-type: none"> • <i>Host</i>: Ligado à massa; • <i>Slave</i>: Não ligado; 	-	-
GND	Massa	-	-
SHELLx	-	-	-

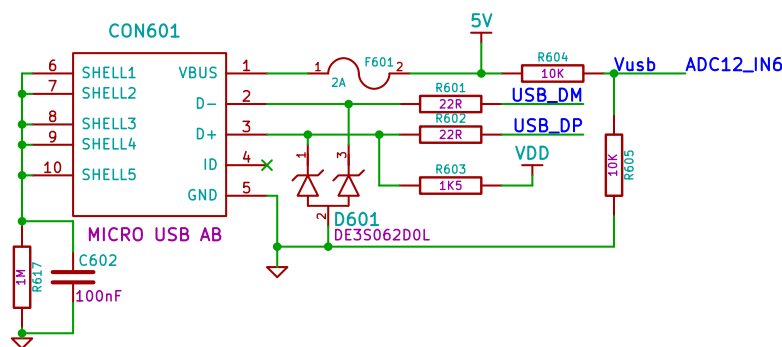


Figura 4.18 – Circuito de ligação da ficha *Micro-USB* ao μC

4.3.5 - Ligação para módulo de alimentação

Este conector serve para fazer a *interface* com o módulo da fonte de alimentação. É constituído pelos pinos de entrada e saída de corrente, com dois níveis de tensão (5V e 3,3V), e a massa. No entanto, também estão disponíveis pinos de entrada/saída para obter informação da fonte de alimentação e poder também atuar sobre a mesma. Para o efeito estão disponíveis seis pinos, dos quais dois são para o barramento de comunicação I²C, uma entrada analógica e três E/S digitais, que estão ligadas ao CI U701 MCP23008. As funções específicas atribuídas a cada um deles, bem como onde esses pinos estão ligados nos respetivos CI, estão presentes na tabela 4.11.

Tabela 4.11 – Lista das Correspondências entre o interface 'Power_Conn' e os diversos CI do Módulo

Power_Conn			Destino		
Nome	Nº	Função	Nome	Nome Pino	Num. Pino
Vcc	1,2	Alimentação de 3,3V	-	-	-
Vss	3,4	Massa	-	-	-
5V	5,6	Alimentação de 5V	-	-	-
I2C1_SCL	7	Linha de relógio do I ² C	μC	PB6	92
I2C1_SDA	8	Linha de dados do I ² C	μC	PB7	93
V _{bat}	9	Tensão da bateria	μC	PB0	35
PWR_GPIO0	10	Controlo do modo de funcionamento do Conversor DC-DC	U701	GPIO1	11
PWR_GPIO1	11	Estado do carregador de baterias (a carregar ou não)	U701	GPIO2	12
PWR_GPIO2	12	Ativação do carregamento das baterias	U701	GPIO3	13

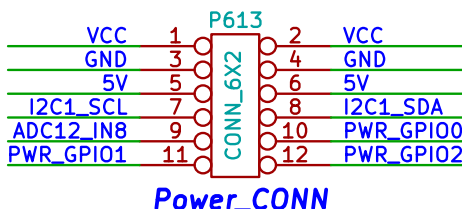


Figura 4.19 – Circuito de ligação do conector 'Power_CONN'

4.3.6 - Ligação para comunicação Wi-Fi

Este conector serve para fazer a *interface* com o módulo de comunicação com o Wi-Fi. O módulo escolhido comunica com o μC através de uma *UART*. Assim, este conector é constituído pelos pinos

de alimentação, dois pinos para a troca de informação com o módulo Wi-Fi (TX e RX da USART3 do μC), dois pinos para controlo de fluxo (RTS e CTS da USART3 do μC) e quatro E/S disponíveis para realizar um controlo mais eficaz do módulo.

As funções específicas atribuídas a cada um deles, bem como onde esses pinos estão ligados nos respetivos CI, está presente na tabela 4.11, e o esquema do conector na figura 4.20.

Tabela 4.12 – Lista das Correspondências entre o interface 'WIFLY_Conn' e os diversos CI do Módulo.

WIFLY_Conn			Destino		
Nome	Nº	Função	Nome	Nome Pino	Num. Pino
Vcc	1	Alimentação de 3,3V	-	-	-
Vss	2	Massa	-	-	-
USART3_TX	3	Pino de transmissão de dados da UART3	μC	PD8	55
USART3_RX	4	Pino de receção de dados da UART3	μC	PD9	56
USART3_CTS	5	Pino de CTS de dados da UART3	μC	PD11	58
USART3_RTS	6	Pino de RTS de dados da UART3	μC	PD12	59
WIFLY_RST	7	Pino de <i>Reset</i> do Módulo <i>WiFly</i>	μC	PD13	60
WIFLY_ASS_STATUS	8	Notificação que o módulo está ligado a uma rede <i>Wi-Fi</i> e com IP válido	μC	PD14	61
WIFLY_TCP_CONN_TRIG	9	Controlo da ligação TCP ao servidor remoto (pino a 1 ativa ligação, pino a 0 fecha ligação)	μC	PD15	62
WIFLY_TCP_CONN_STATUS	10	Notificação que o módulo tem uma ligação TCP ativa.	μC	PC13	7

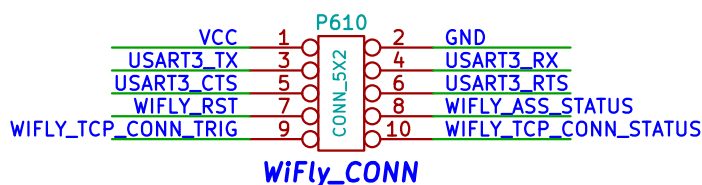


Figura 4.20 – Circuito de ligação do conector 'WiFly_CONN'.

4.3.7 - Ligação para módulo de controlo de tampas

Este conector serve para fazer a *interface* com o módulo de controlo das tampas. O módulo foi concebido para controlar 35 cargas (atuadores dos trincos dos compartimentos) de uma forma matricial para poupar pinos de controlo ao μC . Para controlar essas 35 cargas, são necessários 12 pinos dos quais sete são para controlar as colunas e cinco para controlar as linhas. A estes 12 pinos, como não poderia deixar de ser, é necessário acrescentar mais dois para a alimentação do módulo.

As funções específicas atribuídas a cada um deles, bem como onde esses pinos estão ligados nos respetivos CI, está presente na tabela 4.13, e o esquema do conector na figura 4.21.

Tabela 4.13 – Lista das Correspondências entre o interface 'Ctrl_tampas_Conn' e os diversos CI do Módulo.

Ctrl_tampas_Conn		Função	Destino		
Nome	Nº		Nome	Nome Pino	Num. Pino
VCC	1	Alimentação de 3,3V	-	-	-
VSS	2	Massa	-	-	-
TAMPA_L0	3	Ativação/Desativação da Linha 0	μC	PA5	30
TAMPA_L1	4	Ativação/Desativação da Linha 1	μC	PB1	54
TAMPA_L2	5	Ativação/Desativação da Linha 2	μC	PB2	37
TAMPA_L3	6	Ativação/Desativação da Linha 3	μC	PE7	38
TAMPA_L4	7	Ativação/Desativação da Linha 4	μC	PE8	39
TAMPA_C0	8	Ativação/Desativação da Coluna 0	μC	PE9	40
TAMPA_C1	9	Ativação/Desativação da Coluna 1	μC	PE10	41
TAMPA_C2	10	Ativação/Desativação da Coluna 2	μC	PE11	42
TAMPA_C3	11	Ativação/Desativação da Coluna 3	μC	PE12	43
TAMPA_C4	12	Ativação/Desativação da Coluna 4	μC	PE13	44
TAMPA_C5	13	Ativação/Desativação da Coluna 5	μC	PE14	45
TAMPA_C6	14	Ativação/Desativação da Coluna 6	μC	PE15	46

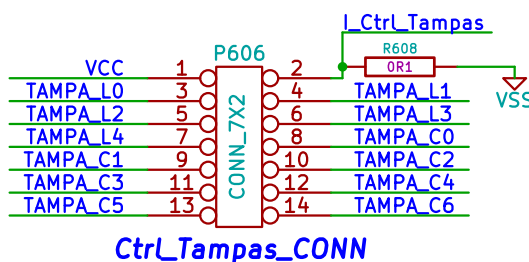


Figura 4.21 – Circuito de ligação do conector 'Ctrl_Tampas_CONN'

4.3.8 - Ligação para módulo de interface Ethernet

Este μC possui um periférico de *Ethernet* que lhe permite enviar e receber dados pela *Ethernet* em conformidade com o *standard* IEEE 802.3-2002, necessitando apenas de um CI externo de acesso ao meio físico, também conhecida, na gíria, com PHY, para que este fique habilitado a funcionar. O interface com o PHY pode ser feito recorrendo a uma de duas *interfaces standard* neste meio:

1. O *Media Independent Interface* (MII);
2. O *Reduced Media Independent Interface* (RMII), sendo este último o escolhido por ocupar menos pinos do μC . [37, p. 1099].

O PHY escolhido foi o DP83848C pelas seguintes razões: É compatível com o μC ; tem disponibilidade de *drivers* e aplicações para o μC ; tem disponibilidade do componente em PCI pronto a funcionar. Apesar de haver outros CI com um custo mais reduzido, o facto de não haver *drivers* para o μC é muito importante, já que reduz o tempo e a complexidade de desenvolvimento do *firmware*.

O posicionamento dos pinos na ficha foi feito de forma a que esta seja eletricamente compatível com a placa "DP83848-Ethernet-Board" da WaveShare. As funções específicas atribuídas a cada um deles, bem como onde esses pinos estão ligados nos respetivos CI, estão representados na tabela 4.14; o esquema do conector está representado na figura 4.22.

Tabela 4.14 – Lista das Correspondências entre o interface 'ETH_Conn' e os diversos CI do Módulo.

ETH_Conn		Função	Destino		
Nome	Nº		Nome	Nome Pino	Num. Pino
VCC	1,2	Alimentação de 3,3V	-	-	-
VSS	3,4	Massa	-	-	-
ETH_RMII_MDIO	5	Management Data I/O	μ C	PA2	25
ETH_RMII_MDC	6	Management Data Clock	μ C	PC1	8
ETH_RMII_REF_CLK	7	RMII Reference Clock	μ C	PA1	24
ETH_RMII_CRS_DV	8	RMII Carrier Sense/Receive Data Valid	μ C	PA7	32
ETH_RMII_RXD0	9	RMII Receive Data 0	μ C	PC4	33
ETH_RMII_RXD1	10	RMII Receive Data 1	μ C	PC5	34
ETH_RMII_TX_EN	11	RMII Transmit Enable	μ C	PB11	48
ETH_RMII_TXD0	12	RMII Transmit Data 0	μ C	PB12	51
ETH_RMII_TXD1	13	RMII Transmit Data 1	μ C	PB13	52
ETH_RMII_TX_CLK	14	MII Transmit Clock	μ C	PC13	7

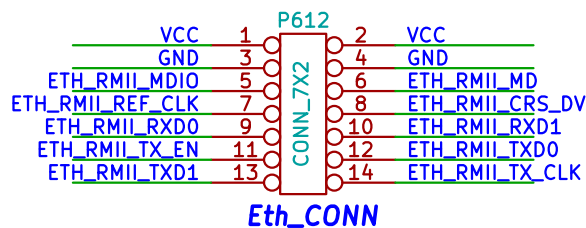


Figura 4.22 – Circuito de ligação do conector 'Eth_CON'.

4.3.9 - Ligação para o módulo de comunicação GSM

Foi também incluída uma *interface* de comunicação com o módulo GSM. Assim, na ligação GSM_CONN, estão disponíveis os pinos de transferência de dados de uma UART (Tx e Rx). Também estão disponíveis pinos para controlo de fluxo (RTS e CTS) que podem ser utilizados com GPIO normais para outras funções. Por fim, também foram incluídos mais dois pinos de utilização geral. As funções específicas atribuídas a cada um deles, bem como onde esses pinos estão ligados nos respetivos CI, está presente na tabela 4.15, e o esquema do conector está representado na figura 4.23.

Tabela 4.15 – Lista das Correspondências entre o interface 'GSM_Conn' e os diversos CI do Módulo.

GSM_Conn		Função	Destino		
Nome	Nº		Nome	Nome Pino	Num. Pino
Vcc	1	Alimentação de 3,3V	-	-	-
Vss	2	Massa	-	-	-
USART2_TX	3	Pino de transmissão de dados da UART2	μ C	PD5	86
USART2_RX	4	Pino de receção de dados da UART2	μ C	PD6	87
USART2_CTS	5	Pino de CTS de dados da UART2	μ C	PD3	84
USART2_RTS	6	Pino de RTS de dados da UART2	μ C	PD4	85
GSM_GPIO0	7	E/S de utilização genérica	μ C	PD0	81
GSM_GPIO1	8	E/S de utilização genérica	μ C	PD1	82

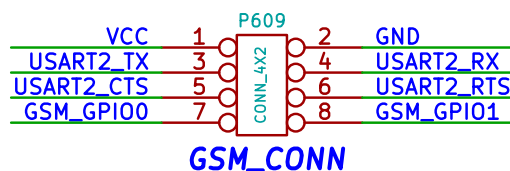


Figura 4.23 – Circuito de ligação do conector 'GSM_CON'

4.3.10 - Ligação para uma mini-coluna de áudio

Como já havia sido referenciado anteriormente, o dispositivo tem que ter um aviso sonoro. Para conseguir esse objetivo este tem que incluir uma mini-coluna que permite ao dispositivo emitir sons. Como as colunas consomem mais de 20mA (corrente máxima que os pinos do μC consegue fornecer), foi incluído um circuito de *interface* para poder comutar uma corrente superior. Para tal foi utilizado um MOSFET (Q601) para fazer a comutação da mini-coluna. A resistência R616 é utilizada para manter a *gate* do Q601 com 0V enquanto o pino que o controla (PE6) estiver em modo de alta impedância. O esquema pode ser visto na figura 4.24.

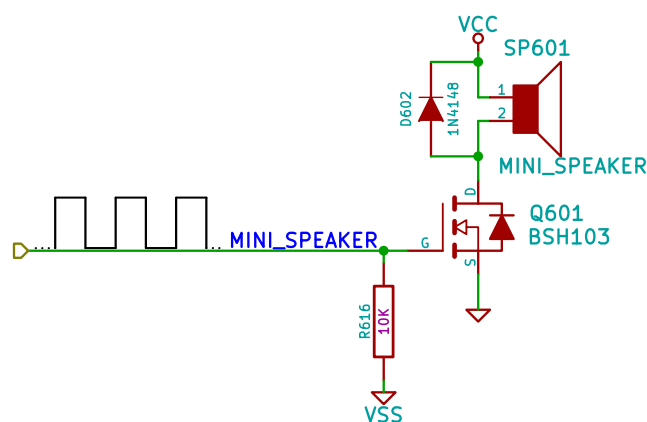


Figura 4.24 – Circuito de ligação da mini-coluna presente no dispositivo.

4.3.11 - Ligação para Entradas e Saídas Externas

Este conector tem por função controlar outros componentes que não podem ser colocados em nenhuma PCI, uma vez que têm que ser colocados em locais estratégicos, como é o caso dos botões de interação com o utilizador, sinalizadores, etc...

Os primeiros pinos são os GPIOs do μC e os restantes são do U701. É necessário ter em atenção que, quando o μC está em modo de *standby*, todos os GPIOs ficam em modo de alta-impedância (com algumas exceções) e não conseguem acordar o μC . Esta última limitação impossibilita a ligação de entradas que tenham que ser continuamente monitorizadas. Também foi adicionado um conjunto de pinos com massa e Vcc.

As funções específicas atribuídas a cada um deles, bem como onde esses pinos estão ligados nos respetivos CI, está presente na tabela 4.16 e o esquema do conector está representado na figura 4.25.

Tabela 4.16 – Lista das Correspondência entre o interface 'I/O_Conn' e os diversos CI do Módulo

I/O_Conn			Destino		
Nome	Nº	Função	Nome	Nome Pino	Num. Pino
PE0	1	Entrada/Saída digital Nº0	μC	PE0	97
PE1	2	Entrada/Saída digital Nº1	μC	PE1	98
PE2	3	Entrada/Saída digital Nº2	μC	PE2	1
PE3	4	Entrada/Saída digital Nº3	μC	PE3	2
I/O_Exp_GPIO7	5	Botão de toma	U701	GP7	17
I/O_Exp_GPIO6	6	Botão de Pânico	U701	GP6	16
I/O_Exp_GPIO5	7	Sensor de detecção da gaveta das tomas	U701	GP5	15
I/O_Exp_GPIO4	8	Sensor de porta de tomas aberta	U701	GP4	14

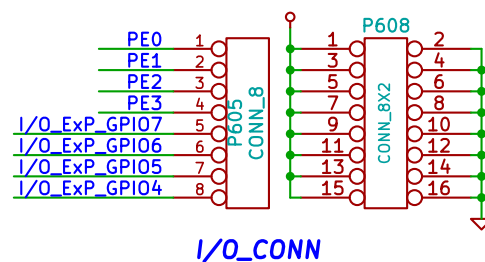


Figura 4.25 – Circuito de ligação do conector 'I/O_CON'

4.3.12 - Ligação para o módulo de áudio

Nesta *interface* foram colocados os pinos necessários para realizar a *interface* com o CI de descodificação de áudio e/ou DAC com amplificação. Utilizando estes CI é possível fazer com que o dispositivo consiga reproduzir sons mais elaborados, como, por exemplo, música, voz, etc... Os sons são fornecidos ao μC sob a forma de ficheiros armazenados no cartão MicroSD.

As funções específicas atribuídas a cada um dos pinos, bem como onde se encontram ligados nos respetivos CI, estão presente na tabela 4.17, e o esquema do conector está na figura 4.26.

Tabela 4.17 – Lista das Correspondências entre o interface 'Audio_Conn' e os diversos CI do Módulo.

Audio_Conn			Destino		
Nome	Nº	Função	Nome	Nome Pino	Nº Pino
VCC	1,2	Alimentação de 3,3V	---	---	---
VSS	3,4	Massa	---	---	---
I2S2ext_SD	5	I ² S – Pino extra para com. <i>full duplex</i> – mapeado no MISO-SPI	μC	PB14	53
I2S2_MCK	6	I ² S – Master Clock	μC	PC6	63
I2S2_SD	7	I ² S – Pino <i>Serial Data</i> – mapeado no MOSI-SPI	μC	PB15	54
Audio_DREQ	8	Pedido de interrupção do Módulo de áudio	μC	PC0	15
I2S2_CK	9	I ² S – Linha de Relógio para o Módulo de áudio	μC	PB10	47
Audio_RST	10	<i>Reset</i> do Módulo de áudio	μC	PB4	90
I2S2_WS	11	I ² S – Word Select – mapeado no pino NSS do SPI	μC	PB9	96
Audio_xDCS	12	<i>Data Chip Select</i> do Módulo de áudio	μC	PC2	17
I2C1_SDA	13	Pino de dados do Barramento I ² C	μC	PB7	93
I2C1_SCL	14	Pino de Relógio do Barramento I ² C	μC	PB8	95

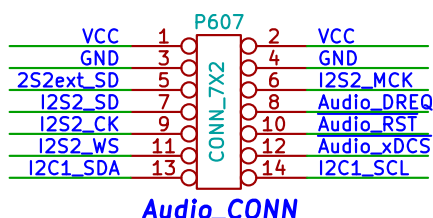


Figura 4.26 – Circuito de ligação do conector 'Audio_CONN'.

4.3.13 - Ligação para de comunicação *Bluetooth*

Foi também incluída uma *interface* para comunicação com o módulo *Bluetooth*. Assim, na ligação BT_CONN estão disponíveis os pinos de transferência de dados de uma UART (Tx e Rx). Também estão disponíveis mais dois pinos de utilização geral que podem ser utilizados para as mais diversas tarefas.

As funções específicas atribuídas a cada um dos pinos, bem como onde estes estão ligados nos respetivos CI, estão presentes na tabela 4.18 e o esquema do conector está representado na figura 4.27.

Tabela 4.18 – Lista das Correspondência entre o interface BT_Conn' e os diversos CI do Módulo

GSM_Conn Nome	Nº	Função	Destino		
			Nome	Nome Pino	Num. Pino
Vcc	1	Alimentação de 3,3V	---	---	---
Vss	2	Massa	---	---	---
USART1_TX	3	Pino de transmissão de dados da UART1	μ C	PA9	68
USART1_RX	4	Pino de receção de dados da UART1	μ C	PA10	69
---	5	---	---	---	---
---	6	---	---	---	---
BT_GPIO0	7	E/S de utilização genérica	μ C	PA8	67
BT_GPIO1	8	E/S de utilização genérica	μ C	PD10	57

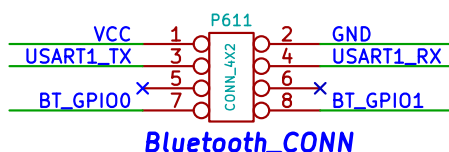


Figura 4.27 – Circuito de ligação do conector 'BT_CONN'.

4.4 - Módulo de controlo dos atuadores das tampas

O Módulo de controlo dos atuadores das tampas tem como função, como o próprio nome indica, controlar os atuadores de abertura dos compartimentos do tabuleiro de tomas, o qual é constituído por 35 compartimentos, cada um deles com o seu respetivo atuador.

4.4.1 - Os atuadores das tampas

Conforme pode ser visto mais adiante no ponto 7.3.2 - da página 86 - é necessário um atuador para abrir eletronicamente os compartimentos dos medicamentos. Depois de alguma pesquisa encontrou-se um dispositivo capaz de satisfazer as necessidades. O atuador escolhido foi um fio chamado *nitinol* que também é conhecido como músculo eletrónico, ou ainda *flexinol*.

4.4.1.1 - O nitinol

O nitinol é baseado numa liga metálica de 50% Níquel-Titânio, desenvolvida pelo US Naval Ordnance Laboratory, e daí o seu nome [42, p. 122]. Apesar de haver outros que já haviam conseguido produzir ligas com propriedades semelhantes, esta entidade foi a primeira a conseguir produzi-la com recurso a materiais não tóxicos [43, p. 50]. Esta liga é da família das ligas com memória de forma.

As ligas com memória de forma possuem uma estrutura cristalina que pode ser facilmente deformada quando está abaixo da temperatura de transição sem se danificar. Quando a liga é aquecida, a estrutura cristalina do material volta a alinhar-se e o material volta à forma anterior à deformação. Desde que não se excedam alguns limites, tais como a temperatura e a tensão de deformação, o material suporta muitos ciclos sem perder a forma [44]. Na figura 4.28 pode-se ver o comportamento atrás descrito.

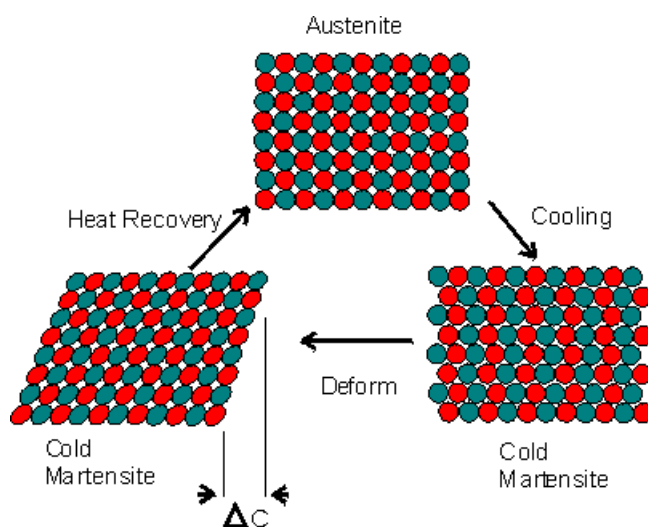


Figura 4.28 – Alteração da estrutura cristalina do *nitinol* durante o processo de deformação, aquecimento e arrefecimento [44].

4.4.1.2 - A aplicação

O *nitinol* pode ter vários formatos, mas para a sua aplicação no presente projeto, o mais adequado é o de fio fino e comprido, também designado de fio músculo, por se comportar com um músculo, ou ainda por *flexinol*¹². Aproveitando esta funcionalidade pode ser utilizado na forma ilustrada na figura 4.29, onde se pode ver a diferença entre o comprimento (ΔL) do fio relaxado e contraído.

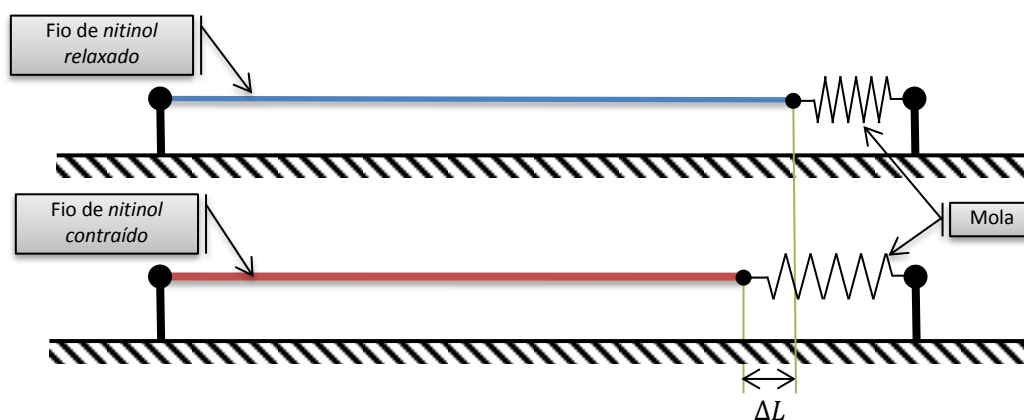


Figura 4.29 – Exemplo de aplicação do *nitinol*.

¹² Os termos *Flexinol*® e *Muscle Wires*® são marcas registadas da *Dynalloy, Inc.*

Como a contração e o relaxamento só dependem da temperatura, qualquer método de aquecimento e arrefecimento podem ser utilizados. Como a liga é má condutora elétrica (cerca de $55\Omega/m$ para um fio com um diâmetro de $150\mu m$ [45, p. 6]), fazendo-lhe passar uma corrente elétrica, por efeito de joule, facilmente se pode aquecer o material até à temperatura de transição. O arrefecimento dá-se cortando a corrente elétrica e deixando-o dissipar o calor para o ambiente circundante. Ao arrefecer, e com a ajuda da mola, o fio volta à posição inicial.

A escolha do diâmetro a utilizar foi feita com base em dois parâmetros: força de contração e corrente necessária para o ativar. Com base na tabela 4.30, de [45, p. 6], o diâmetro escolhido foi de $150\mu m$, com uma temperatura de transição de 70° e possui as seguintes características apresentadas na figura 4.30.

Flexinol	
150μm	
Flexinol Wire Properties	
Type:	Flexinol 150
Diameter:	150 μm
Activation Temp:	70 °C 90°C
Length:	1 metre
Resistance:	50 Ω/m
Recommended Current	400mA
Rec. Recovery Force:	330gmf
Rec Deformation Force	62gmf
Rec Deformation:	4-5%

Figura 4.30 – Lista de características do *flexinol*® utilizado.

4.4.1.3 - Modelização

A modelização deste atuador não é difícil, já que este se comporta, em termos elétricos, como uma resistência. Depois de discutir com a restante equipa do projeto, chegou-se à conclusão que um deslocamento máximo (ΔL_{MAX}) de $0,5cm$ seria suficiente para o mecanismo de abertura dos compartimentos dos medicamentos. Atendo a que às características referidas acima serão necessários cerca de $10cm$ de fio, que corresponde a uma resistência aproximada de 5Ω e a uma corrente máxima recomendada de $400mA$.

4.4.2 - Projeto e conceção do módulo

Tendo em consideração o que foi descrito anteriormente seriam necessários 35 pinos de saída do μC com mais 35 MOSFET, dado que a corrente consumida pelos atuadores é maior do que o máximo permitido em cada pino. Isto não seria viável dado que:

- Seria necessário alocar mais de 35% dos pinos do μC para esta função, ficando a faltar para outras aplicações;
- A quantidade de condutores de interligação das placas (Módulo do μC com o de controlo dos atuadores das tampas) seria impraticável dado a distância a que as placas vão ficar umas das outras;
- A quantidade de componentes necessários para controlar cada um dos atuadores individualmente faria encarecer o equipamento e também obrigaria a que as placas fossem maiores ocupando assim mais espaço no equipamento.

Tendo em consideração estes inconvenientes, e sabendo que só é necessário atuar um atuador de cada vez, optou-se por uma abordagem diferente. O controlo dos atuadores foi feito sob a forma matricial, como é comum fazer num teclado ou numa matriz de LED, para citar apenas alguns exemplos.

Controlo Direto

- 35 GPIOs;
- 37 Condutores entre módulos:
 - 35 para controlo ;
 - 2 para alimentação;
- 35 Drives de potencia (Conjunto de MOSFET e resistências de polarização);



Controlo Matricial

- 12 GPIOs necessárias:
 - 5 para controlar linhas;
 - 7 para controlar as colunas;
- 12 Drives de potencia (Conjunto de MOSFET e resistências de polarização);
 - 5 para controlar linhas;
 - 7 para controlar as colunas;

Figura 4.31 – Comparação entre os recursos necessários para controlo direto e controlo matricial.

Como se pode ver na figura 4.31 há uma poupança significativa em termos de recursos utilizados, com a única limitação de não se poder ligar os atuadores de qualquer forma. No entanto, dado o funcionamento de equipamento, só é necessário ativar um compartimento de cada vez, logo, não se torna uma limitação.

Para facilitar *design* na conceção do *driver* (MOSFET e resistência de polarização) foram colocados na PCI o *nitinol* que estará fisicamente nos compartimentos, e a sua ligação será feita no tabuleiro das tomas. Só as linhas e colunas é que irão ligar à PCI.

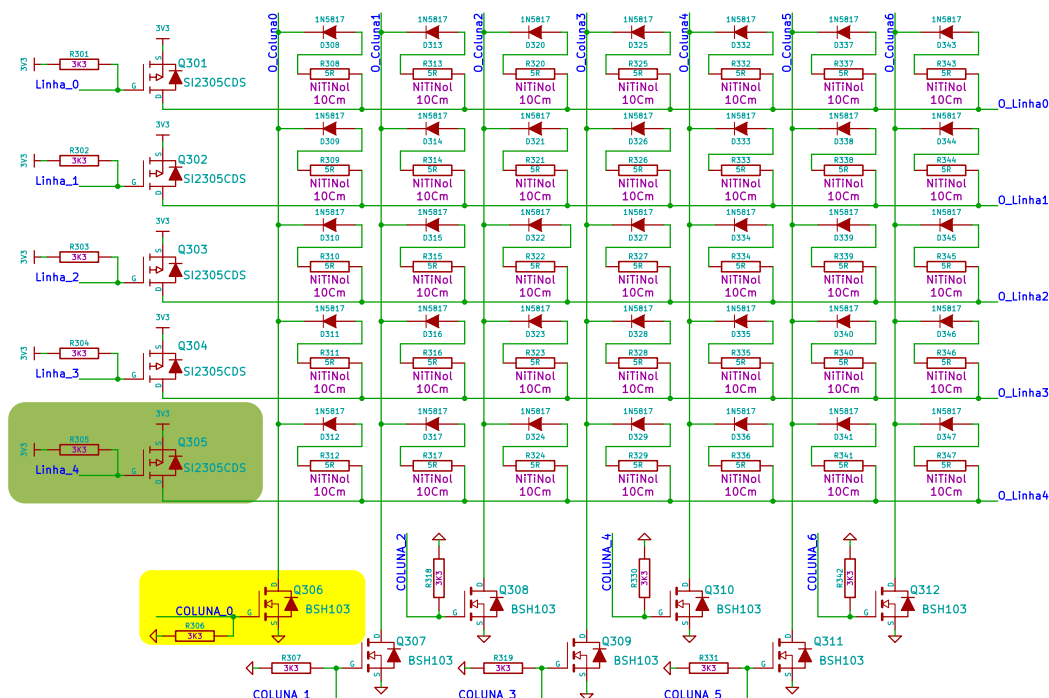


Figura 4.32 – Esquema da matriz de controlo dos atuadores das portas dos compartimentos.

Na figura 4.32 está representado o esquema da matriz de controlo dos atuadores. Dentro do retângulo verde está o *driver* de potência da linha quatro. Este é constituído por um MOSFET de canal P (SI2305CDS); a resistência R305 serve para manter o canal do MOSFET fechado mesmo que o pino do μC fique em modo de alta impedância, o que acontece quando este está no modo mais profundo de hibernação. No retângulo amarelo está presente o *driver* da coluna zero. Este é muito semelhante ao *driver* da linha, mas utilizando um MOSFET de canal N (BSH103); a resistência R306 serve para evitar que a *gate* do MOSFET fique a flutuar quando o pino de controlo ficar em modo de alta impedância.

Para ativar qualquer um dos atuadores é necessário ativar uma linha e uma coluna. Por exemplo, se se quiser ativar o R323, é necessário ativar a linha três e a coluna dois.

Para que este sistema funcione com resistências é necessário colocar em série com cada uma das resistências um diodo para evitar que a corrente flua no sentido errado. Neste caso, a corrente deverá fluir das linhas para as colunas (considerando o sentido convencional da corrente). Caso não se coloque, ao ativar uma linha e uma coluna, ao invés da corrente fluir por uma só carga, fluirá por diversas cargas, o que aumentará o consumo sem necessidade, e fará com que várias cargas possam ser ativadas acidentalmente. Na figura seguinte estão assinalados os percursos da corrente ao ser ativada uma determinada linha e uma coluna.

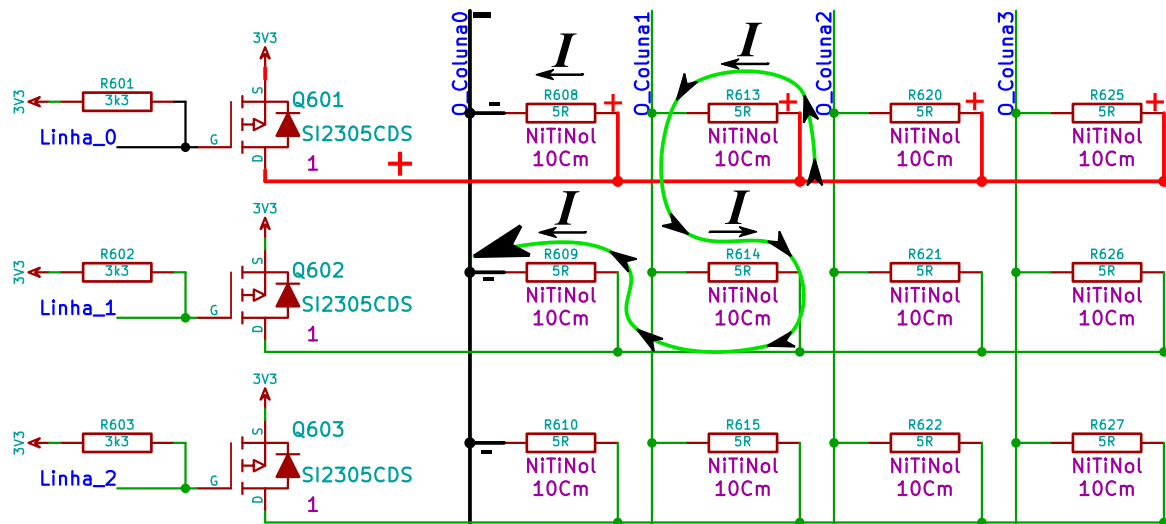


Figura 4.33 – Demonstração da passagem de corrente por diversas cargas quando estão ativas a linha e coluna 0 (no caso da ausência dos diodos de controlo do fluxo de corrente).

Como se pode observar na

figura 4.33, apesar de só se querer ativar a resistência R608, o que acontece é que há corrente a passar por todas as resistências da linha 0, o que não deve acontecer.

De facto, há resistências em que a corrente, que deveria passar exclusivamente da linha para a coluna, deixam passar corrente no sentido contrário. Para evitar que tal aconteça foi necessário adicionar um diodo em série com a respetiva carga, já que este componente só permite que a corrente passe num único sentido.

Para além deste aspeto, poderá ser necessário adicionar uma resistência para limitar a corrente no *nitinol*. Assim, em série com o *nitinol* é necessário colocar uma resistência para que, quando ativado, a corrente que o atravessa não exceda os 400mA.

4.5 - Módulos de comunicação

Está previsto que este equipamento possa comunicar através de diversos meios. No entanto, apesar de estar previsto apenas um meio de comunicação por *ePiilBox*, esta poderá levar os quatro ao mesmo tempo. A escolha poderá ser feita na altura da montagem ou da aquisição.

4.5.1 - Módulo Wi-Fi

O coração do módulo Wi-Fi é o CI RN-171 que, por sua vez, está montado num suporte compatível com o módulo *Xbee*. Este módulo é constituído pelo RN171 e por uma PCI compatível com *Xbee*, à qual se dá o nome de RN-171-VX-WiFly [46]. Este módulo tem as seguintes funcionalidades:

- Velocidade máxima de transmissão de informação de 460kbps através de UART;

- Gestão de alimentação inteligente com eventos de acordar programáveis (temporizador e E/S);
- Relógio de tempo real para *timestamp*, *auto-sleep* e vários modos de *auto-wakeup*;
- Configuração através de *Wi-Fi* ou por *UART* usando comandos simples em ASCII;
- *Upgrade* de *firmware* “via ar” (*over the air*) através de FTP;
- Autenticação de *Wi-Fi* segura através de: WEP, WPA-PSK (TKIP), WPA2-PSK;
- Aplicações de *networking*: DHCP, DNS, ARP, ICMP, UDP, Telnet, FTP e cliente HTML;
- Poupança de energia em modo de 802.11 b/g e funções de *roaming*;
- Potência de transmissão ajustável de: -2dBm a 12dBm ;
- Modo de configuração WPS para uma configuração mais fácil e segura;
- Cliente *web* HTML interno para enviar as GPIO, UART e dados dos sensores para um servidor *web* remoto.

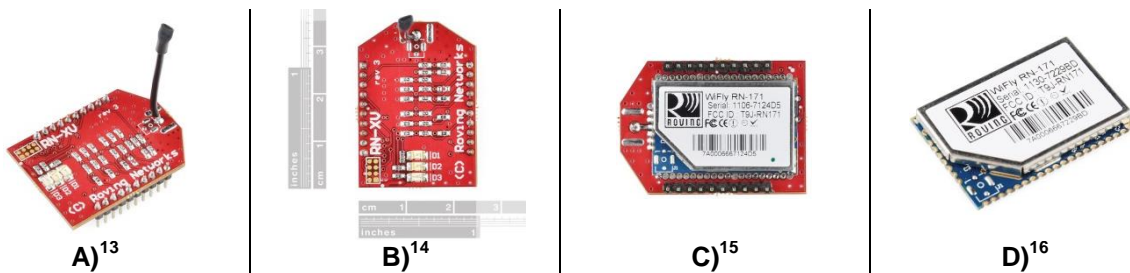


Figura 4.34 – Fotografia dos módulos “RN-XV WiFly”(A,B,C) e do “RN-171”(D).

Na figura 4.34 estão presentes várias imagens do módulo utilizado. A utilização do módulo RN-171-XV WiFly tem vantagens já que é compatível com o formato *Xbee* [46, p. 1] que utiliza o *standard* 802.15.4 (também conhecido por *ZigBee*). A norma 802.15.4 é uma norma de comunicação sem fios de baixo consumo. A utilização do módulo RN-171 tem vantagens por ser um pouco mais barato e mais compacto e, por isso, ocupar menos espaço. O módulo escolhido para integrar o dispositivo nesta fase de protótipo será o “RN-171-XV WiFly” por ser mais simples de utilizar fisicamente e o seu custo extra não ser significativo.

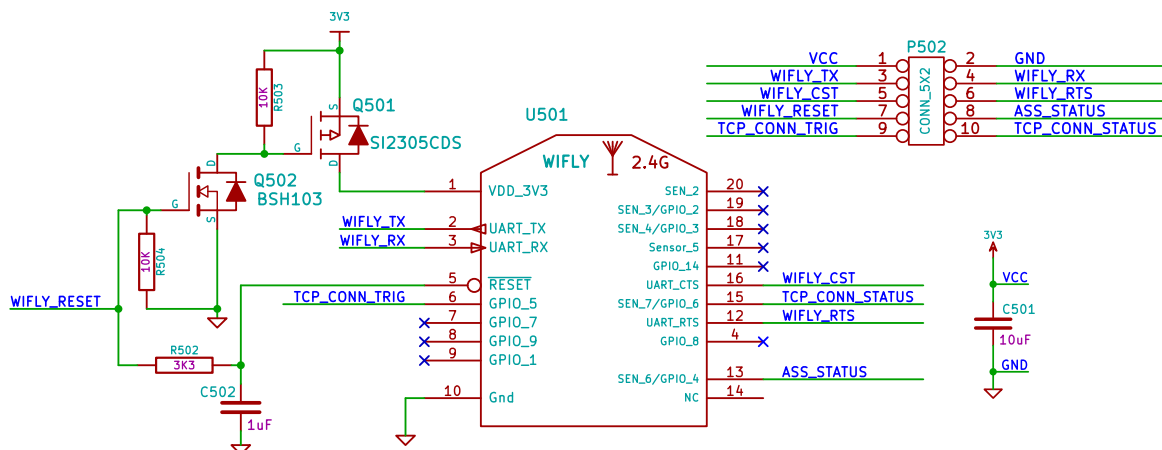


Figura 4.35 – Circuito do módulo de comunicação com o Wi-Fi.

Na figura 4.35 ilustra o circuito eletrónico do módulo, podendo observar e pode-se ver que o pino “WiFly_RST” tem a dupla função de cortar a alimentação ao módulo e, simultaneamente, controlar o *reset* do módulo.

¹³ Fonte: <https://dlnmh9ip6v2uc.cloudfront.net/images/products/1/0/8/2/2/10822-01.jpg>;

¹⁴ Fonte: <https://dlnmh9ip6v2uc.cloudfront.net/images/products/1/0/8/2/2/10822-02.jpg>;

¹⁵ Fonte: <https://dlnmh9ip6v2uc.cloudfront.net/images/products/1/0/8/2/2/10822-03.jpg>;

¹⁶ Fonte: <https://dlnmh9ip6v2uc.cloudfront.net/images/products/1/1/0/4/9/11049-01.jpg>;

No anexo E estão indicados os procedimentos para parametrizar o módulo Wi-Fi para que este possa interagir corretamente com o μC .

4.5.2 - GSM

Este módulo tem como função ao *interface* com o *modem* GSM da MULTITECH modelo MTSMC-G-F1, sendo alimentado com uma tensão de 5V. Dado que o μC funciona com 3,3V há a necessidade de realizar uma adaptação de sinais entre o μC e o modem [47].

As principais características do módulo GSM são as seguintes:

- GPRS Classe 10;
- Dual-band 850/1900 ou 900/1800 GSM/GPRS;
- GSM Classe 1 e Classe 2 e Grupo 3 FAX;
- Serviço de mensagens curtas;
- 14.4K GSM circuit switched data;
- Conector de antena MMCX e suporte para cartão SIM;
- *Interface* série suporta velocidades até 115.2K;
- Compatível com comandos AT;
- Compressão de dados V.42bis;
- Gestão de lista telefónica na memória e no SIM.

A nível de consumo, este *modem* absorve, tem uma corrente máxima de 1,7A durante as rajadas de transmissão [47, p. 9].

4.5.2.1 - Adaptador de níveis de tensão

A forma encontrada para fazer adaptação dos sinais foi através do circuito da figura 4.36:

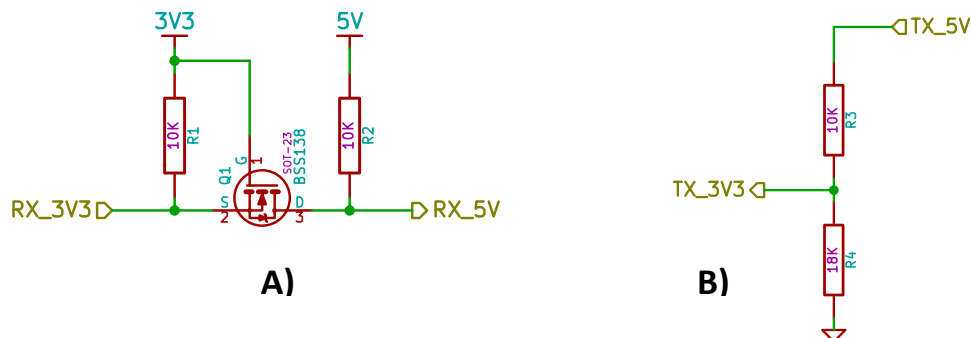


Figura 4.36 – Circuito de adaptação de níveis de sinal.

A figura 4.36 a) ilustra um circuito que permite a adaptação de sinais bidirecionais; o circuito da figura 4.36 b) permite adaptar sinais que provenham do lado da tensão mais elevada para o lado da tensão mais baixa [48, pp. 9-11].

O funcionamento do circuito a) é o seguinte:

- Quando o dispositivo, com um nível de tensão baixo (RX_3V3), transmite um '1' (3,3V), a tensão V_{GS} é igual a 0V o que faz com que o MOSFET (Q_1) mantenha o canal fechado, e o lado com um nível de tensão mais elevado (RX_5V) veja 5V através da resistência de *pull-up* R_2 . Quando a linha RX_3V3 transmite um '0' (0V), a *source* do MOSFET é aterrado, o que faz com que o valor de V_{GS} seja 3,3V, e o MOSFET abra o canal fazendo com que o lado de mais alta tensão seja puxado para 0V;
- Quando o lado de mais alta tensão transmite um '0' (0V), o diodo do substrato do MOSFET conduz, puxando do lado de mais baixa tensão para aproximadamente 0,7V; esta tensão é suficientemente baixa para abrir o canal do MOSFET, puxando o lado de mais baixa tensão para 0V, ou quase. Quando esta linha transmite um '1' (5V), o pino *source* do MOSFET é

puxado até 3,3V e o MOSFET fecha o canal fazendo com que apareça 3,3V, ou seja, '1' no lado de mais baixa tensão através da resistência de *pull-up* R_1 .

O funcionamento do circuito B) é mais simples e pode ser considerado um divisor de tensão, já que os pinos dos CI envolvidos consomem uma corrente muito baixa. Usando a fórmula do divisor de tensão presente em (4.1), e substituindo pelos valores de tensão e resistência utilizados neste circuito, tem-se:

$$V_{TX_3V3} = \frac{R_4}{R_3 + R_4} \times V_{TX_5V} \Rightarrow \frac{18k}{10k + 18k} \times 5 = \frac{18}{28} \times 5 \cong 3,21V \quad (4.9)$$

Quando é aplicado o valor lógico '1' (5V) no lado de mais alta tensão (TX_5V), devido ao circuito em causa, o valor de tensão que vai aparecer no lado de tensão mais baixa (TX_3V3) é de aproximadamente 3,21V. Este valor de tensão é percebido como sendo '1'. Quando em TX_5V é aplicado o valor lógico de '0' (0V) a tensão em TX_3V3 será também de 0V, ou seja, '0'.

4.5.2.2 - Controlo de Alimentação e Reset

Para evitar consumos desnecessários, quando o módulo GSM não estiver a ser utilizado foi implementado, à semelhança do módulo de *Wi-Fi*, um circuito de corte de alimentação. Na figura 4.37 pode-se ver no retângulo laranja (b) o circuito que controla a alimentação do módulo. Quando o pino GSM_PWR_CTRL for colocado a '1' (3,3V) o módulo MTSMC-G-F1 fica alimentado. Se colocado a '0' (0V), desabilita a alimentação do módulo. O Q804 alimenta o módulo GSM quando é aplicado um sinal de 0V na sua *gate*, para tal acontecer é necessário aplicar um sinal positivo na *gate* do Q805. A resistência R811 serve para manter o V_{GS} de Q804 com uma tensão igual zero e assim manter o canal fechado, enquanto Q805 está com o canal fechado. A resistência R812 serve para manter a *gate* de Q805 com um valor de tensão zero e assim manter o canal fechado, mesmo que não seja aplicada nenhuma tensão na sua *gate*.

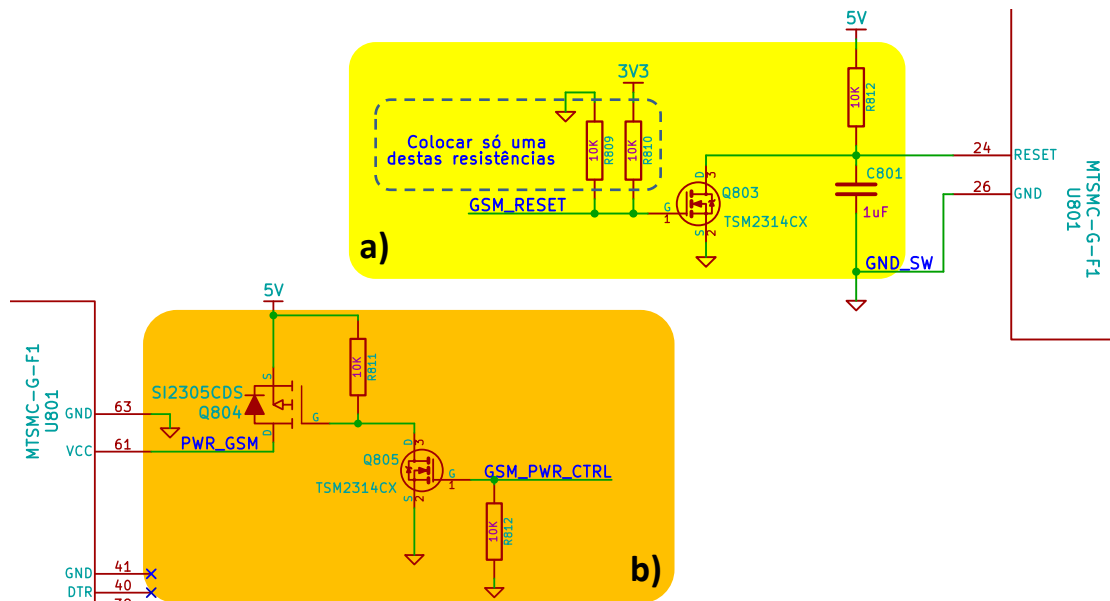


Figura 4.37 – Circuito de controlo de alimentação e Reset do Módulo MTSMC-G-F1.

O circuito dentro do retângulo amarelo (a) contém o circuito de *reset* do módulo. A resistência R812 de *pull-up* serve para colocar o pino de *reset* a nível alto, desativando o *reset* e colocando o módulo em funcionamento. O Q803 foi colocado devido à diferença de tensão entre o módulo e o μC , respetivamente 5V e 3,3V. As resistências R809 e R810 não podem ser colocadas ao mesmo tempo na placa – estas resistências têm como função definir o estado do *reset*, caso não seja aplicada tensão nenhuma no pino GSM_RESET. Caso se pretenda que o *reset* esteja ativado por omissão

deve-se colocar a R810, caso contrário deve-se colocar a R809, que fará com que o *reset* seja desativado por defeito.

4.5.2.3 - Circuito do módulo

Devido à dimensão deste esquema foi colocado no anexo B, juntamente com todos os restantes esquemas do projeto.

4.5.3 - Bluetooth

O módulo Bluetooth utilizado foi o HC-06, também conhecido como 'Linvor', da *Guangzhou HC Information Technology Co., Ltd.* Este módulo possui duas variedades, um mestre e outro escravo. O módulo mestre tem a capacidade de se ligar a um ou vários escravos, mas não se pode ligar a outro módulo mestre. O módulo escravo só pode comunicar com módulos mestre e outros dispositivos com *Bluetooth*, como é o caso de PC e *smartphones*, etc.

O módulo em causa é bastante simples e funciona com uma UART sem fios. Permite assim enviar informação do μC para o dispositivo *Bluetooth* de forma transparente. As características do dispositivo são [49]:

- Transmissor Sem fios:
 - Possuir uma sensibilidade (*Bit error rate*) que pode chegar aos $-80dBm$;
 - A gama de potências de transmissão vai de: $-46dBm$ a $+6dBm$.
- Descrição das funcionalidades:
 - Tem módulo EDR –Pode alterar a taxa de modulação entre *2Mbps* e *3Mbps*;
 - Incorpora antena interna de *2.4GHz*;
 - Possui uma *flash* externa de *8Mbits*;
 - Tem uma gama de tensões de alimentação ente *3,1V* até *4,2V*;
 - O consumo de corrente varia entre *30mA* e *40mA* durante o emparelhamento e *8mA* durante as restantes comunicações;
 - Possui uma HCI Standard (UART);
 - Possui formato de SMD;
 - É produzido pelo processo RoHS;
 - Possui um tamanho de: $27mm \times 13mm \times 2mm$;
 - Tem um circuito externo com poucos componentes;
 - Suporta uma gama de temperatura de funcionamento entre $-25^{\circ}C$ e $+75^{\circ}C$.

Na figura 4.38, dentro do retângulo amarelo está o circuito de *reset* do módulo. A resistência R303 de *pull-up* serve para colocar o pino de *reset* a um nível alto, desativando o *reset* e colocando o módulo em funcionamento, caso não seja aplicada nenhuma tensão em 'BT_RESET'. Caso seja aplicado *0V* em 'BT_RESET', o módulo deixará de funcionar. O circuito dentro do retângulo laranja (b) controla a alimentação do módulo Bluetooth. O seu funcionamento é bastante simples bastando aplicar *0V* em 'BT_PWR_CTRL' para habilitar a alimentação do módulo e *3,3V* para a inibir.

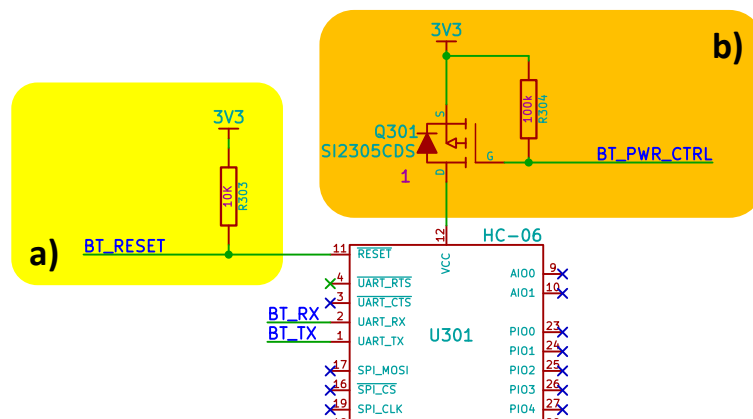


Figura 4.38 – Circuito de controlo de alimentação e Reset do Módulo HC-06.

Para facilitar o desenvolvimento do protótipo, e dado que o módulo *Bluetooth* não era prioritário, este não foi desenvolvido em PCI. Em substituição, foi adquirida uma solução com base no HC-06, mas montada numa PCI e com alguns componentes que garantem o seu correto funcionamento (*BT_Board V1.4*). Para o interligar com a placa do μC foi necessário fazer um cabo para adaptar as ligações de um módulo para o outro. Na figura 4.39 pode ver-se uma fotografia do módulo e as restantes ligações ao módulo do μC .

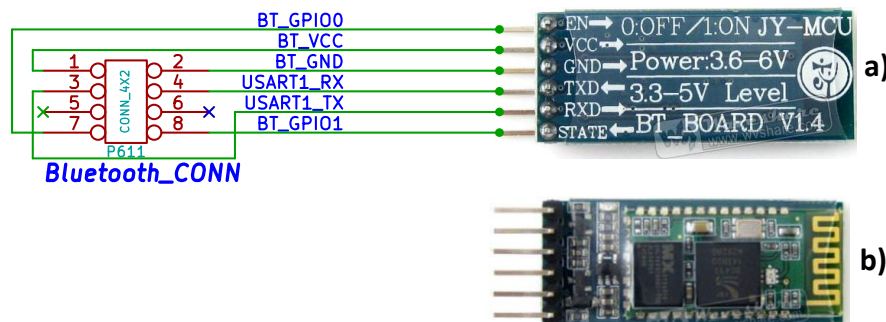


Figura 4.39 – Foto do Módulo BT_BOARD V1.4 e ligação à placa do módulo μC ¹⁷.

4.5.4 - Ethernet

O μC possui, implementado por *hardware*, o Controlo de Acesso ao Meio ou, em inglês, *Media Access Control* (MAC), que faz parte da camada 2 (Enlace de dados) do modelo OSI. Para ter acesso à *Ethernet* necessita que algo lhe faculte o acesso ao meio (camada 1 do modelo OSI) ou, em inglês, *Physical Layer*. O CI escolhido para colocar no projeto foi o DP83848C. As razões da escolha são as seguintes:

- Compatível com o μC ;
- Disponibilidade de *drivers* e aplicações para o μC ;
- Disponibilidade do componente em PCI pronto a funcionar.

Existem outros CI com um custo mais reduzido; no entanto, o facto de não haver *drivers* para o μC , aumenta o tempo e complexidade de desenvolvimento do *firmware*, tornando-os assim numa solução mais dispendiosa.

Para o protótipo, em vez de se desenvolver uma PCI específica, foi utilizada uma placa comercial da “*Waveshare Electronics*”, que pode ser vista na figura 4.40. As ligações entre os módulos são

¹⁷ a) fonte: <http://www.wavshare.com/img/devkit/accBoard/Bluetooth-Slave-UART-Board/Bluetooth-Slave-UART-Board-4.jpg>

b) fonte: <http://www.wavshare.com/img/devkit/accBoard/Bluetooth-Slave-UART-Board/Bluetooth-Slave-UART-Board-2.jpg>

asseguradas por um cabo paralelo. O conector ‘ETH_CONN’ tem os pinos nas mesmas posições que o conector da PCI para facilitar a interligação entre elas.

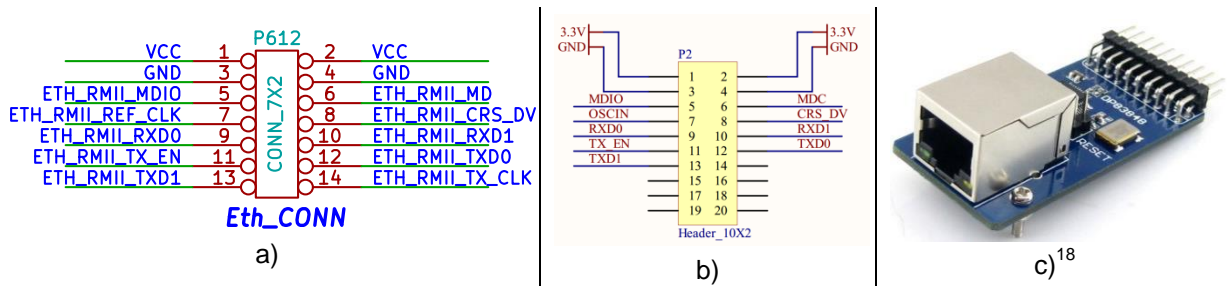


Figura 4.40 – a): Circuito de ligação do conector ‘Eth_CONN’; b): Circuito de ligação do conector da placa “DP83848-Ethernet-Board” [50]; c): Imagem da “DP83848-Ethernet-Board”.

4.6 - Desenho da placa de circuito impresso

A conceção de placas de circuito impresso (PCI) é um processo minucioso e de elevada complexidade, sujeito a muitas restrições e condicionalismos. Estes obrigam ao seguimento de regras apertadas para evitar erros que provoquem o mau funcionamento da PCI, mesmo que não haja erros de ligações.

Apesar de se tratar de um protótipo, o seu objetivo é que venha um dia a ser produzido em massa e comercializado. Tendo isso em mente foi necessário haver um compromisso entre a funcionalidade requerida pelo protótipo e o custo associado. Nesse sentido as PCI foram desenvolvidas de forma modular permitindo realizar alterações módulo a módulo. Também foram utilizados componentes SMD permitindo reduzir o tamanho das placas e facilitando a automatização da montagem. Sendo um protótipo e com as ferramentas de soldadura disponíveis, foram escolhidos componentes SMD não muito pequenos.

4.6.1 - Normas e regras a respeitar

Tal como na maior parte das áreas, existem normas que governam o projeto de PCI. Das pesquisas efetuadas, o conjunto de normas IPC-2220, desenvolvidas pelo “*Institute for Interconnecting and Packaging Electronic Circuits*”, estabelecem os requisitos para o desenho de placas de circuito impresso, estando organizadas mediante a figura 4.41. [51]

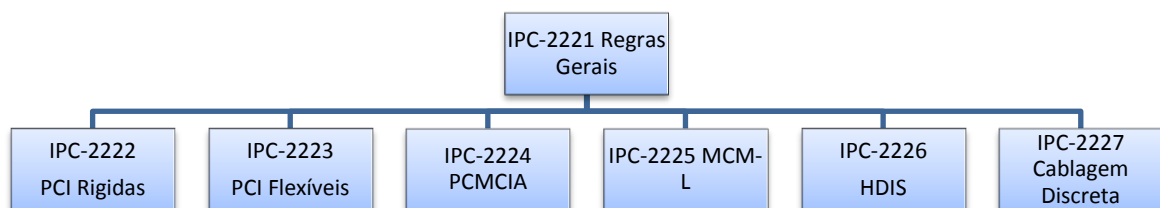


Figura 4.41 – Hierarquia das especificações de *design* (Serie 2220).

A norma aplicável para esta PCI seria a IPC-2221 e a IPC-2222. Não foi possível obter uma cópia da mesma por falta de verbas para a sua aquisição. No entanto existem muitos guias de boas práticas, ou simples recomendações, que são fornecidas através de pesquisas minuciosas na internet, principalmente em alguns fabricantes de PCI, [52] e [53].

Com base na experiência adquirida e nas referências apresentadas anteriormente, apresenta-se de seguida uma lista com algumas recomendações e regras a cumprir para um bom desenho de uma placa de circuito impresso:

¹⁸ c) fonte: <http://www.wvshare.com/img/devkit/accBoard/conn-open/DP83848-Ethernet-Board-1.jpg>

- As pistas devem ser o mais curtas possível e devem-se evitar ângulos retos na mesma camada, para não criar picos de corrente ou capacidades parasitas;
- Ter cuidado na colocação das vias evitando a concentração das mesmas num local para não fragilizar demasiado a PCI. Também não se devem colocar demasiadamente próximas das ilhas dos componentes;
- Se a PCI for para produzir em massa devem-se escolher componentes de montagem de superfície em detrimento dos outros. Os componentes de montagem de superfície, ou SMD, facilitam a montagem das PCI com recurso a maquinaria especializada. Este requisito é imprescindível para produção em massa.
- É aconselhável deixar algum espaço sem componentes (2 a 5mm) de, pelo menos, 2 lados da PCI. Este procedimento tem o objetivo de facilitar a montagem e soldadura automatizada dos componentes, já que esse espaço é utilizado para suporte da PCI;
- Devem ser colocados condensadores de desacoplamento, nas fontes de tensão e nos CI (conforme as recomendações do fabricante), entre o positivo e a massa para filtrar as componentes de tensão de alta frequência;
- Os conetores e pistas que contenham sinais analógicos devem ser afastados, o mais possível, de fontes de ruído, como é o caso das linhas de alimentação. Este procedimento tem como objetivo evitar as interferências e aumentar a fidelidade do valor analógico medido;
- A distribuição dos componentes deve ser uniforme pela placa, salvo exceções, como são o caso das zonas de alívio térmico, etc.. Esta distribuição permite que os componentes possam dissipar melhor o calor produzido e não aquecer demasiado;
- Colocar todas as conexões com o exterior da PCI nas periferias, se possível no mesmo lado. Isto facilita a conceção do invólucro e a respetiva montagem;
- Os componentes devem ser colocados alinhados uns com os outros, e na vertical ou horizontal. Se a PCI for retangular, os CI, por exemplo, devem ser colocados segundo a mesma orientação. Devem ficar alinhados entre si, se tal não impedir a criação de pistas mais curtas e retas;
- Deve-se consultar as especificações do fabricante a fim de aferir se não se está a violar nenhuma das suas recomendações. Isto é especialmente válido no caso de componentes que emitam radiofrequências ou produzam muito calor. Caso não se respeitem as normas do fabricante corre-se o risco da PCI não funcionar convenientemente, apesar de, aparentemente, estar tudo bem interligado;
- Deve-se distribuir os componentes que produzem muito calor por diversas zonas da PCI para evitar a concentração de calor numa zona específica;
- Apesar das ferramentas de desenho de PCI estarem cada vez mais evoluídas, o roteamento automático destas ferramentas deve ser sempre sujeito a verificação humana e aos parâmetros do fabricante da PCI.

4.6.2 - Especificações técnicas

As especificações técnicas da PCI dizem respeito ao número de camadas que constituem a PCI, largura e isolamento das pistas, diâmetro das vias e respetivo espaçamento mínimo, e espessura mínima das linhas da serigrafia. A maioria destes parâmetros, principalmente os mínimos, é determinada pelos fabricantes de PCI.

As placas foram desenvolvidas com duas camadas, uma superior e outra inferior, sendo que os componentes devem ser colocados, de preferência, na face superior. A utilização de apenas duas camadas faz com que o custo de produção da placa seja mais baixo, apesar das quatro camadas facilitarem o desenho das pistas.

A largura das pistas varia consoante a corrente máxima que normalmente fluirá nelas. Assim, a grande maioria das pistas terão uma espessura mínima, já que a corrente que lá irá passar é mínima. A largura padrão das pistas é de 0,254mm (10mil). As pistas de alimentação, que são as que vão ter que suportar uma corrente superior (cerca de 1A em regime contínuo), serão as mais grossas, com 1,2mm (47mil) de largura. O isolamento entre pistas será de 0,2mm (8mil). Estas larguras foram calculadas considerando que a espessura de cobre utilizada é de 35µm. Os valores de largura da pista e isolamento foram calculados com recurso à ferramenta de *software* “PCB Calculator”, que faz parte da *suite* utilizada para a conceção dos esquemas e PCI. Segundo esta aplicação, os valores por ela calculados estão em conformidade com a norma IPC-2221. Na figura 4.42 a) está apresentado o exemplo dos valores calculados para a largura mínima das pistas de alimentação. Na figura 4.42 b) está apresentada uma tabela com os espaçamentos mínimos entre pistas, mediante a diferença de tensão entre elas.

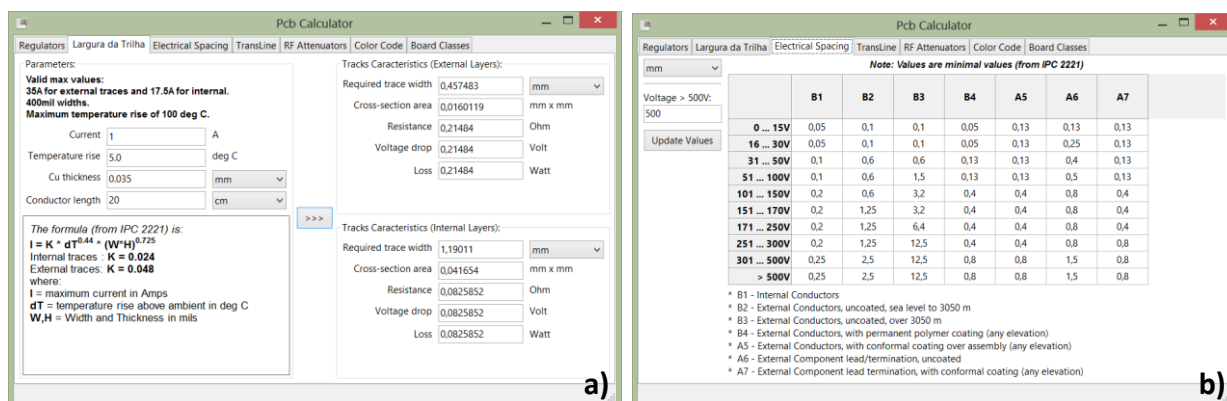


Figura 4.42 – Ecrã do *software* "PCB Calculator"

Finalmente, o diâmetro padrão das vias é de 0,8mm, com um furo de 0,4mm. O diâmetro para as vias, nas pistas de alimentação, é de 1,2mm, com um furo de 0,5mm. O espaçamento mínimo entre vias é igual ao espaçamento entre pistas. Um resumo das especificações das pistas e vias pode ser observado na tabela 4.19.

Tabela 4.19 – Especificações das pistas e vias.

Classe	Isolamento	Largura pista	Diâmetro daVia	Furo da Via
Predefinida	0,2mm	0,254mm	0,8mm	0,4mm
Alimentação de CI	0,2mm	0,580mm	0,8mm	0,4mm
Alimentação	0,25mm	1,2mm	1,2mm	0,5mm

4.6.3 - Disposição de componentes

Utilizando os valores definidos nas especificações técnicas, o “Pcbnew” (aplicação com a função de desenhar PCI na *suite* KiCad), permite desenhar a PCI respeitando os parâmetros definidos. Ao contrário de outras *suites* de desenho de PCI, este *software*, não possui roteador automático. Para compensar este facto há a possibilidade de se utilizar um roteador automático externo, designado por FreeROUTE¹⁹. Como seria de esperar, este *software* fica aquém de *suites* pagas, como é o caso do OrCAD ou do EAGLE²⁰. No entanto, é um bom ponto de partida para fazer as primeiras ligações da PCI.

¹⁹ O *Software FreeRoute* é um *software* escrito em java e corre diretamente no browser sem necessidade de efetuar instalação. O endereço onde pode ser encontrado é: <http://www.freerouting.net/>.

²⁰ O EAGLE tem uma versão grátis, mas tem limitações a nível de captura de esquemático e de tamanho de PCI que pode desenhar. E também não pode ser utilizado a nível comercial sem o pagamento de uma licença.

Depois de toda a placa desenhada é possível ver a forma final desta. O *software* tem a possibilidade de gerar um modelo 3D da placa com os componentes montados (nem todos os componentes utilizados têm uma imagem 3D disponível para o KiCad). Nas figuras seguintes podem ver-se as PCI dos diversos módulos com os componentes/zonas identificadas.

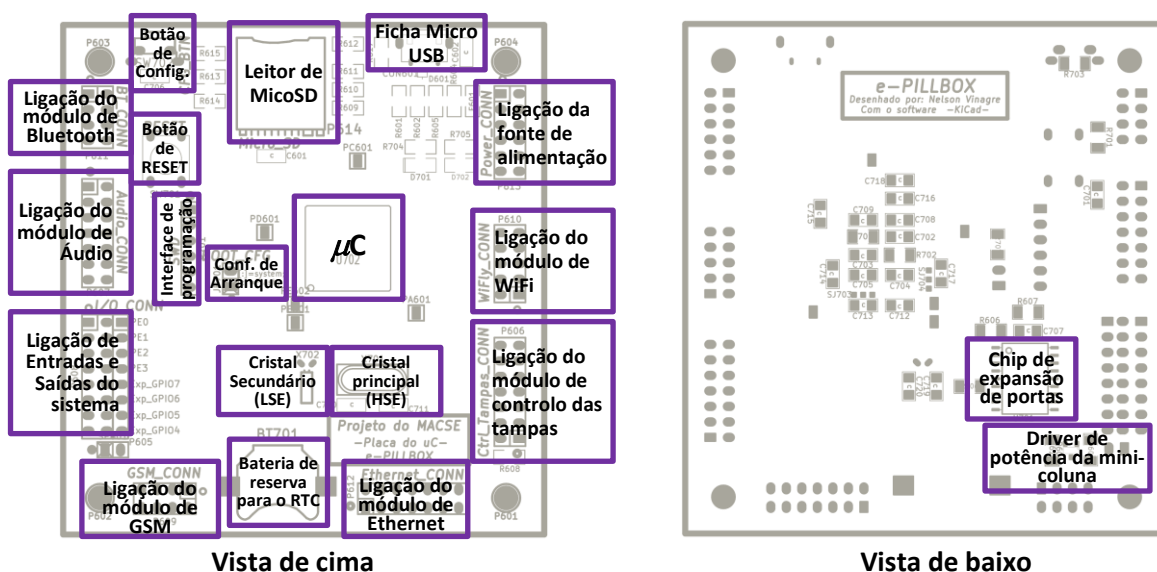


Figura 4.43 – Placa de circuito impresso do Módulo do μ C.

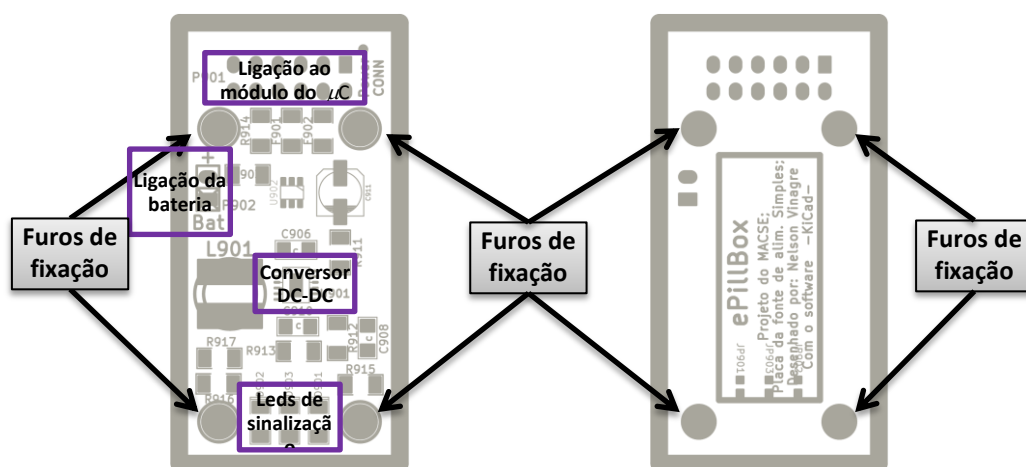


Figura 4.44 – Placa de circuito impresso do Módulo da fonte de alimentação.

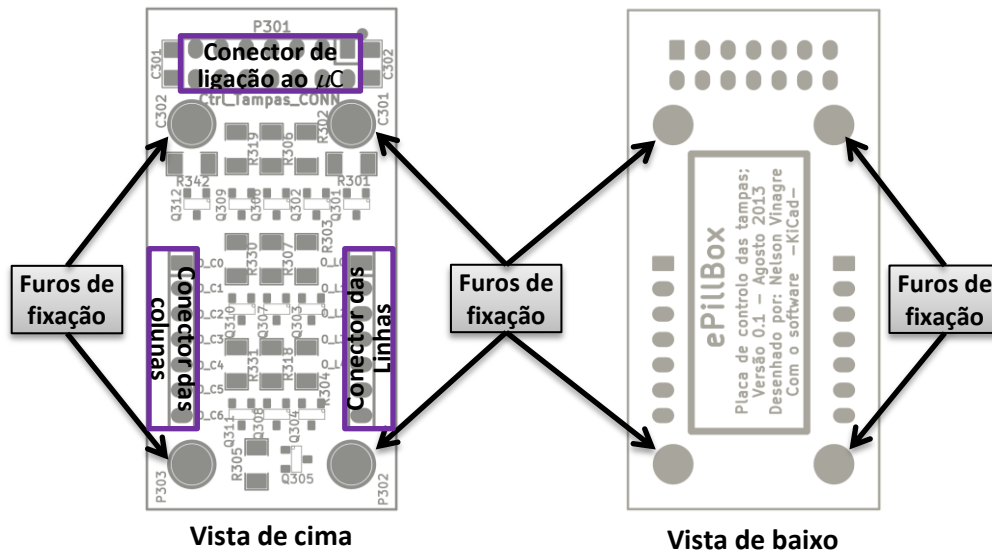


Figura 4.45 – Placa de circuito impresso do Módulo de controlo das tampas.

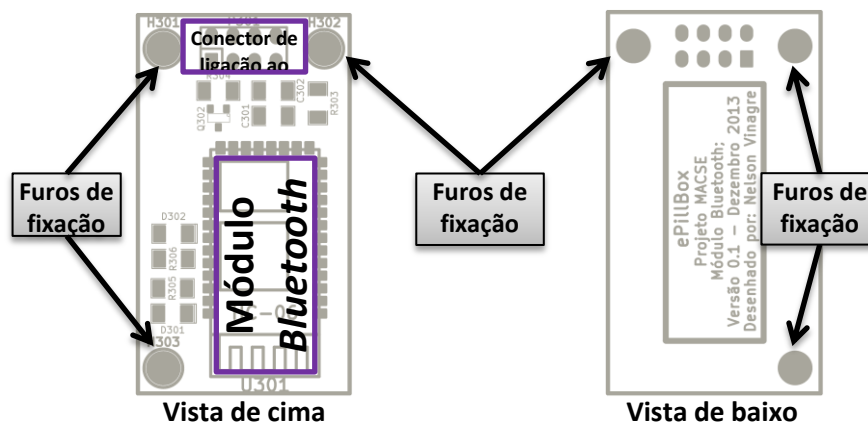


Figura 4.46 – Placa de circuito impresso do Módulo de Bluetooth.

4.6.4 - Planos de massa

Para reduzir as interferências e as capacidades parasitas foram colocados planos de massa nas PCI. O desenho destes planos de massa é automático depois do utilizador delimitar as zonas onde pretende os planos de massa. Depois das zonas delimitadas foi necessário ajustar algumas pistas e colocar algumas vias por forma a aumentar os planos de massa. O aspeto final pode ser visto nas figuras figura 4.47 a figura 4.50:

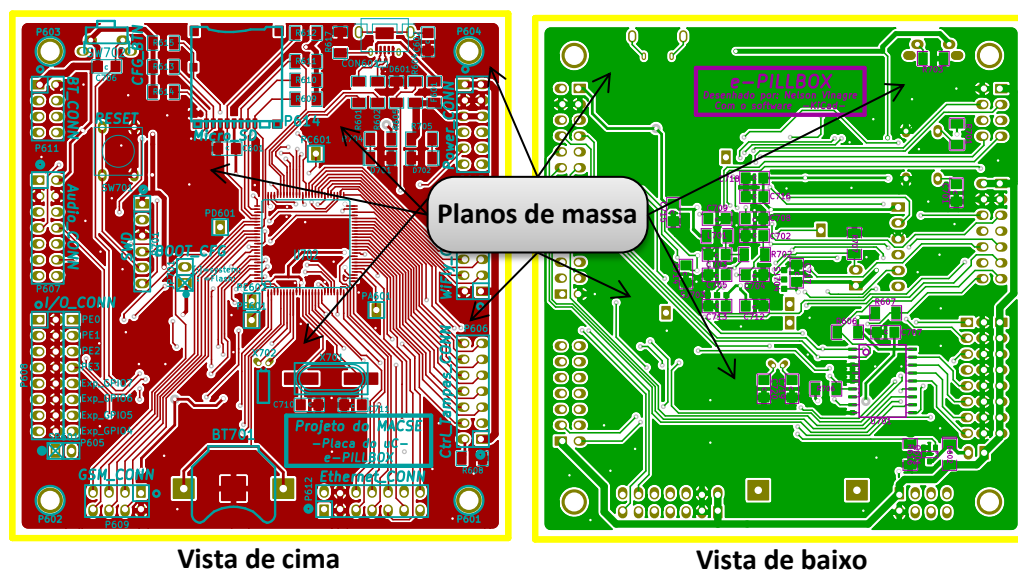


Figura 4.47 – Planos de massa da PCI do Módulo do μ C.

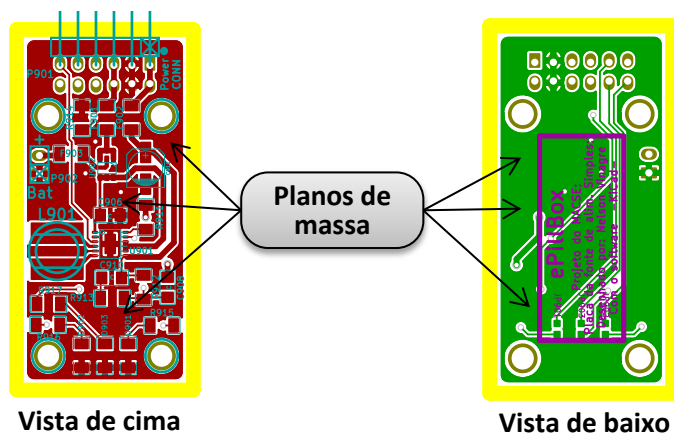


Figura 4.48 – Planos de massa da PCI do Módulo da fonte de alimentação.

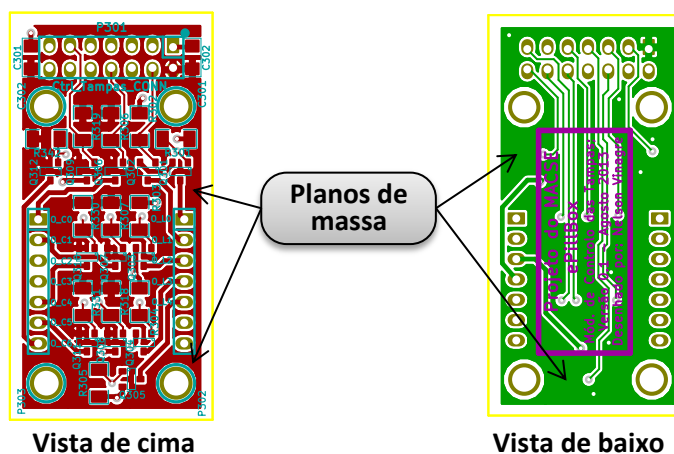


Figura 4.49 – Planos de massa da PCI do Módulo de controlo das tampas.

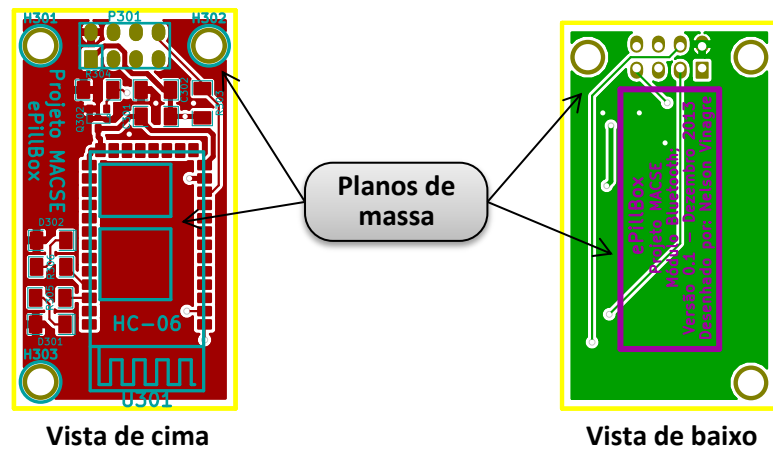


Figura 4.50 – Planos de massa da PCI do Módulo de Bluetooth.

4.6.5 - Legenda da PCI

Por fim, é necessário proceder à identificação dos diversos componentes e terminais de ligação da PCI. Isto permite que se identifique cada componente desenhado no esquemático na PCI facilitando assim a sua operação e reparação. Isto é feito recorrendo ao processo de serigrafia por cima das faces da PCI.

4.7 - Sumário

Neste capítulo foi descrito todo o processo de desenvolvimento de *hardware*. Começando pela elaboração por cada módulo de: especificações; esquemático; definição de regras a seguir no desenho das PCI; criação de planos de massa; e criação das legendas dos componentes e ligações.

Capítulo 5 - Desenvolvimento de software

Este capítulo aborda o desenvolvimento do *software* necessário ao bom funcionamento do projeto. O *software* é uma peça vital para que o restante *hardware* possa desempenhar as funções e cumprir os requisitos descritos na arquitetura do sistema.

5.1 - Enquadramento

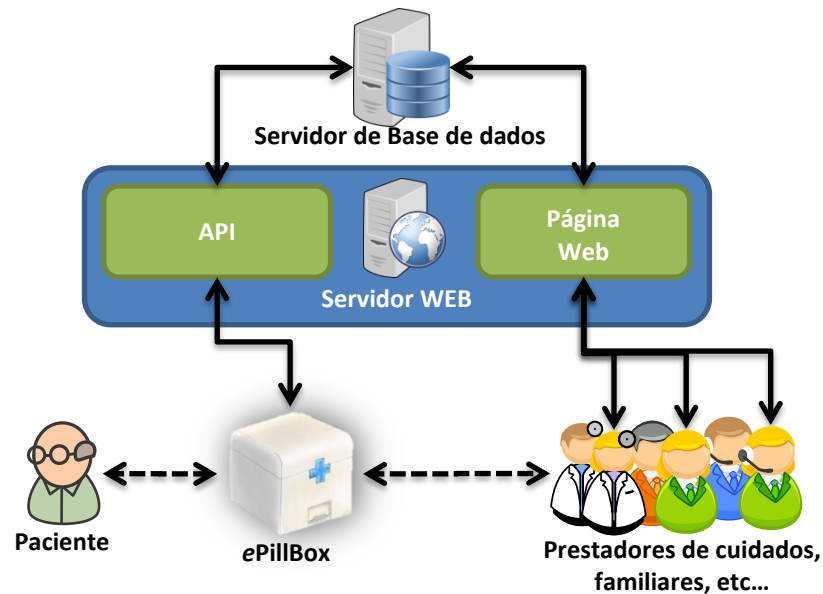


Figura 5.1 – Arquitetura do *software* do sistema.

Na figura 5.1 está presente um diagrama com a arquitetura da forma como os diversos componentes estão organizados e comunicam entre si. Neste caso, o servidor WEB está dividido em duas partes:

1. A página *web* a partir que os diversos utilizadores do sistema utilizam para parametrizar e ver a informação do dispositivo;
2. A API (*Application Programming Interface*) que serve para atender os pedidos vindos dos dispositivos.

A troca de informação entre a API e a página *web* é feita indiretamente através do servidor da base de dados. O servidor da base de dados conterá toda a informação relativa ao sistema. A informação mínima a registar é a seguinte:

- Dados de autenticação dos utilizadores;
- Associação dos dispositivos aos pacientes e aos prestadores de cuidados;
- Registos dos planos de tomas;
- Registo dos parâmetros de cada uma dos dispositivos;
- Registo da conformidade das tomas por parte dos pacientes;
- Outras que venham a ser consideradas importantes.

Por último tem-se o dispositivo – *ePillBox*. O *software* deste dispositivo é responsável por assegurar o funcionamento do *hardware* e comunicar com o servidor WEB para a troca de informação dos horários das tomas, reportar o não cumprimento das tomas e, por fim, as configurações gerais do dispositivo.

Dado a especificidade de cada um dos módulos de *software* utilizados, estes são muito diferentes ao nível da sua conceção, linguagem de programação utilizada e abordagens. Por este motivo, cada um dos programas de *software* será abordado em pontos diferentes.

5.2 - O firmware

O *software* que faz parte do dispositivo também é conhecido como *firmware* – o *firmware*, segundo o IEEE, é a combinação entre memória persistente e *software* guardado nela, (tradução livre de [54]). Este *software* é o que torna o *hardware* digital útil; sem ele o dispositivo não funciona, apesar de fisicamente estar tudo funcional.

O *firmware* foi desenvolvido em linguagem de programação C, utilizando o ‘CooCox CoIDE’, como IDE (*Integrated Development Environment*); este ambiente de desenvolvimento é baseado no conhecido ‘Eclipse IDE’ e permite a edição e a organização, servindo ainda de interface com outras ferramentas que, por sua vez, asseguram a compilação, a programação do μ C e o *debug*. As ferramentas externas são as seguintes:

- **GCC-ARM**²¹ (*GNU Tools for ARM Embedded Processors*) – a ferramenta utilizada para compilar o código fonte em C para código máquina;
- **CoFlash** (da CooCox) – para a programação do μ C;
- **GDB** (*The GNU Project Debugger*) – para fazer o *debug* do *firmware* à medida que vai sendo desenvolvido.

As bibliotecas externas utilizadas no desenvolvimento do *software* foram as seguintes:

- **CMSIS** (*Cortex Microcontroller Software Interface Standard*) – trata-se de uma biblioteca independente de vendedor do μ C que implementa uma camada de abstração de *hardware* para a série de μ C ARM Cortex-M. Esta biblioteca é também suportada por algumas empresas de *software*, nomeadamente em alguns sistemas operativos embebidos e compiladores [23, pp. 164-169]. Isto torna-se vantajoso já que promove a reutilização de código e a portabilidade entre diversos μ C da família ARM Cortex-M, independentemente do vendedor, o que permite uma redução de custos no desenvolvimento de *firmware*.
- **Newlib** – é uma biblioteca ANSI C, desenvolvida para ser implementada em sistemas embebidos. É formada por um conjunto de várias partes de outras bibliotecas, todas com licenças de *software* livre, que as torna facilmente utilizáveis em produtos embebidos, [55];
- **lwIP** – é uma implementação independente do protocolo TCP/IP, que foi inicialmente desenvolvida por Adam Dunkels. O objetivo da lwIP é manter todas as funcionalidades do TCP/IP num sistema de recursos limitados. Isto faz com que seja adequado para sistemas embebidos com dezenas de kBytes de RAM livre e cerca de 40kBytes de memória de programa [56]. Neste projeto será necessária quando o dispositivo estiver ligado à rede Ethernet através do PHY DP83848C.
- **JSMN** – é um analisador minimalista JSON em C, que pode ser facilmente integrado em projetos embebidos com recursos limitados. A sua utilização facilita o trabalho de descodificação das tramas JSON recebidas do servidor WEB, [57].
- **FatFs** – é uma biblioteca que permite a pequenos sistemas embebidos lerem sistemas de ficheiros no formato FAT. A FatFs foi escrita em conformidade com o ANSI C e completamente isolada da camada de acesso ao disco. Esta particularidade torna-a independente de arquitetura de *hardware* [58]. Para complementar a biblioteca é necessário fornecer funções que façam o acesso ao suporte físico através do *hardware* interno do μ C, funções essas que foram obtidas de [59].

²¹ A versão utilizada foi a gcc-arm-none-eabi-4_7-2013q3 e pode ser obtida em: <https://launchpad.net/gcc-arm-embedded>

5.2.1 - O Sistema Operativo de Tempo Real (RTOS)

O sistema operativo (SO) é uma parte opcional do *firmware* de um sistema embebido. O SO pode ser utilizado em qualquer μC , desde que haja uma versão compatível. O SO é constituído por uma série de bibliotecas com dois objetivos, no sistema embebido:

1. Disponibilizar uma camada de abstração para o *software* que venha a ser desenvolvido de forma a ser menos dependente do *hardware*;
2. Gerir os diversos recursos de *software* e *hardware* disponíveis para permitir um funcionamento do sistema o mais eficiente e fiável possível.

Apesar dos SO disponíveis para sistemas embebidos variarem muito a nível dos componentes que possuem, todos incluem um núcleo (*Kernel*), o qual é o componente que possui as funções principais do sistema, como por exemplo [60, pp. 169-171]:

- Gestão de processos/Tarefas;
- Gestão de memória;
- Gestão do sistema de E/S.

Dado que os sistemas embebidos são muito heterógenos, também os SO disponíveis o são, contendo funcionalidades, consumo de recursos e tipos de funcionamento diferentes. Num sistema multitarefa é necessário um mecanismo que determine quando e durante quanto tempo cada tarefa está a correr no CPU – mecanismo chamado de escalonador. Caso o algoritmo do escalonador consiga garantir que a tarefa cumpra com os prazos de execução (*deadline*), e estes sejam previsíveis (determinísticos), então o SO pode ser chamado de *Sistema Operativo de Tempo Real*, ou, em inglês *Real-Time Operating System* (RTOS), [60, p. 199].

5.2.1.1 - Vantagens e desvantagens de um RTOS

A inclusão de um RTOS no projeto facilita o desenvolvimento e ampliação de aplicações de tempo-real. Utilizando corretamente as prioridades de cada uma das tarefas pode-se adicionar tarefas de baixa prioridade que não irão causar impacto significativo nas outras tarefas de prioridade mais elevada. Também permite o desenvolvimento da aplicação dividindo-a em tarefas mais ou menos independentes. Por fim, permite uma utilização mais eficiente dos recursos através de serviços importantes, tais como: semáforos, caixas de correio, filas de espera e temporizadores [60, p. 238].

Como desvantagens tem-se:

- Custo extra do RTOS;
- Necessidade de mais espaço a nível de memória de dados (RAM) e memória de programa (ROM);
- Necessidade de capacidade de processamento extra por parte da CPU;
- Investimento em formação para aprender a utilizar o RTOS.

5.2.1.2 - A escolha do RTOS

A utilização de um RTOS não era obrigatória neste projeto. No entanto, existem vantagens na sua utilização, nomeadamente fazendo com que o desenvolvimento da aplicação seja mais simples já que permite a divisão do projeto em diversas tarefas.

A escolha do RTOS teve em consideração as seguintes características/limitações:

- Não ser proprietário;
- Possuir uma licença sem custos para utilização comercial;
- Suportar as ferramentas de desenvolvimento e o μC utilizados;
- Ser relativamente leve, não consumindo demasiada RAM/ROM ou tempo de processador;
- Ser minimamente estável para ser utilizado num produto final.

Depois de alguma pesquisa e análise [61], o RTOS escolhido foi o “CooCox CoOS”, por se enquadrar nas características anteriormente referidas [62]. Para além do núcleo, também possui serviços de

comunicação e sincronização entre tarefas, tais como, por exemplo: semáforos, caixas de correio, filas de espera, etc. possui ainda módulo de gestão de memória e temporizadores por *software* [63].

5.2.2 - Organização dos blocos de código

O código fonte do *firmware* está organizado por pastas. Cada pasta é composta por vários ficheiros (.c e .h) que correspondem a um conjunto de funções que pertencem a um determinado periférico ou funcionalidade. A organização é a seguinte:

- **CMSIS** – Esta pasta contém os ficheiros da biblioteca CMSIS para o núcleo do microcontrolador em causa (ARM Cortex-M4). Estes ficheiros são fornecidos pela própria ARM e não foram alterados;
- **CMSIS_boot** – Contém os ficheiros de arranque para configurar corretamente os registos internos do μ C. O código é fornecido pelo fabricante do μ C, ou seja, a STM. Neste caso é necessário modificar algumas constantes para que o relógio de sistema seja configurado convenientemente;
- **CMSIS_lib** – Contém as funções específicas para lidar com os diversos periféricos existentes no μ C. Os ficheiros podem ser adicionados ou removidos consoante os periféricos usados;
- **CoOS** – Contém os ficheiros relativos ao RTOS. Neste caso o único ficheiro que deve ser alterado é o *"/Other/OsConfig.h"*. Este ficheiro é o local onde estão as configurações do RTOS;
- **Libs** – Esta pasta contém todas as bibliotecas externas ao projeto, que estão divididas por subpastas:
 - **Jsmn** – Biblioteca utilizada para processamento de *strings* JSON;
 - **Ub_lib** – Esta biblioteca é constituída por duas partes:
 - A biblioteca *FatFs* que permite o acesso a sistemas de ficheiros *FAT*;
 - A implementação do acesso à camada física do μ C por parte da biblioteca *FatFs*;
 - **Wifly_lib** – Esta é uma adaptação, realizada neste projecto, da biblioteca desenvolvida para a plataforma Arduino [64]. A adaptação consistiu na alteração das seguintes funções:
 - Acesso à UART;
 - Alterações da forma como as *strings* utilizadas na comunicação são armazenadas na memória;
 - Adição e adaptação da biblioteca para funcionar com alguns pinos de controlo do dispositivo;
- **Gpio** – Esta pasta contém os ficheiros com as funções que utilizam as GPIO, neste caso:
 - **adc_fun** – Contém as funções de acesso aos ADC, nomeadamente, acesso ao sensor de temperatura interno, as várias tensões do sistema, etc;
 - **ctrl_tampas** – Contém as funções de controlo do módulo das tampas;
 - **speaker** – Contém as funções de controlo da mini-coluna;
- **i2c** – Esta pasta contém as seguintes funções:
 - **i2c** – Acesso ao barramento de comunicação I²C;
 - **mcp23008** – Funções para controlo do CI MCP23008;
- **Shell** – Esta pasta contém a implementação de um interpretador de linha de comandos para poder efetuar alguma interação com o dispositivo. Esta “linha de comandos” só deverá estar disponível para depuração e não para utilização geral;
- **Rtc** – Contém as funções de gestão do módulo de RTCC, que inclui:
 - Configuração;
 - Leitura do tempo e data;
 - Gestão dos alarmes;
 - Gestão das interrupções;
- **Usart** – Esta pasta contém duas versões diferentes do mesmo tipo de funções para acesso às UART do μ C:

1. **usart** – Esta é a versão *standard* de controlo das UART;
2. **usart_shell** – Esta versão de funções é utilizada pela linha de comandos, já que o método de interação com as UART é diferente do normal;
- **Utils** – Esta pasta contém algumas funções de uso genérico, tais como, por exemplo:
 - Funções de temporização em milissegundos e microssegundos; no entanto, estas funções não devem ser utilizadas quando se utiliza um RTOS;
 - Funções de conversão de *strings* em números e vice-versa;
- **WebSvr_comm** – Esta pasta contém as funções de comunicação com o servidor WEB. Nelas são criados os pacotes a serem enviados, é realizada a sua respetiva descodificação e processamento;
- **Bkp_domain** – Esta pasta contém as funções de gestão da memória RAM que está alimentada pela bateria de backup. Esta memória mantém os seus valores mesmo que o sistema não esteja alimentado;
- **Syscalls** – Esta pasta contém as funções de baixo nível para o correto funcionamento de algumas funções da “*new lib C*”.
- **tasks** – Esta pasta contém os diversos ficheiros fonte das tarefas utilizadas para gerir o dispositivo. Existem tarefas específicas, como é o caso da linha de comandos, que são implementadas noutros ficheiros.

Por fim, encontram-se na raiz do projeto alguns ficheiros que contêm as funções de gestão do equipamento, tais como:

- **global_vars** – Aqui estão declaradas todas as variáveis globais que são utilizadas no projeto. Optou-se por esta abordagem para facilitar a gestão das mesmas.
- **ePillBox_Conf** – Este ficheiro contém as configurações específicas do dispositivo *ePillBox*;
- **main** – Este ficheiro contém a função ‘main’, que é a primeira a ser executada depois do μ C efetuar o arranque;
- **stm32f4xx_it** – Este ficheiro contém as implementações das rotinas de serviço às interrupções, principalmente às exceções que são interrupções desencadeadas quando algo de errado ocorre durante a execução do programa;

5.2.3 - Entradas e saída da *ePillBox*

As E/S externas que servirão para o utilizador interagir com o equipamento são as seguintes:

- Entradas:
 - **Botão de libertação de toma** – Este botão tem como função indicar a presença do paciente junto ao equipamento, pois isto garante que este se apercebeu do alarme e está junto do mesmo;
 - **Botão de configuração** – Este botão poderá ter três funções:
 - i. Ativar o modo de configuração do dispositivo;
 - ii. Realizar o *reset* do dispositivo;
 - iii. Realizar a reinicialização do dispositivo;
 - **Botão de pânico** – Servirá para despoletar um pedido ao servidor remoto. Neste caso, a ação a executar será decidida no servidor remoto;
 - **Sensor do compartimento das tomas** – Esta entrada serve para detetar a abertura e o fecho do compartimento das tomas;
 - **Sensor da gaveta de dispensa das tomas** – Este sensor deteta a presença ou não da gaveta das tomas, dado que o local de onde se retira os medicamentos é amovível do equipamento;
- Saídas:
 - **Mini-Coluna** – Serve para efetuar sinalização sonora do dispositivo;
 - **LED de sinalização** – Serve para realizar a sinalização visual do dispositivo. Estes LEDs podem ser colocados todos separados ou juntos. No caso de serem juntos,

num só corpo, podem-se criar todas as cores do arco-íris podendo criar uma solução de sinalização interessante.

- **Vermelho:**
 - **Fixo** – Falha na comunicação com o servidor;
 - **A piscar** – Atingiu a última toma é necessário recarregar;
- **Verde** – A comunicar com sucesso com o servidor;
- **Azul** – Indica a disponibilidade de novo medicamento.

5.2.4 - Descrição do modo de funcionamento do *firmware*.

O *firmware* funciona com base numa máquina de estados. Durante o funcionamento do equipamento o *firmware* vai passando por vários estados, sendo que cada um deles terá uma função específica. Também existem algumas funções que são executadas com uma determinada periodicidade, independentemente do estado do dispositivo.

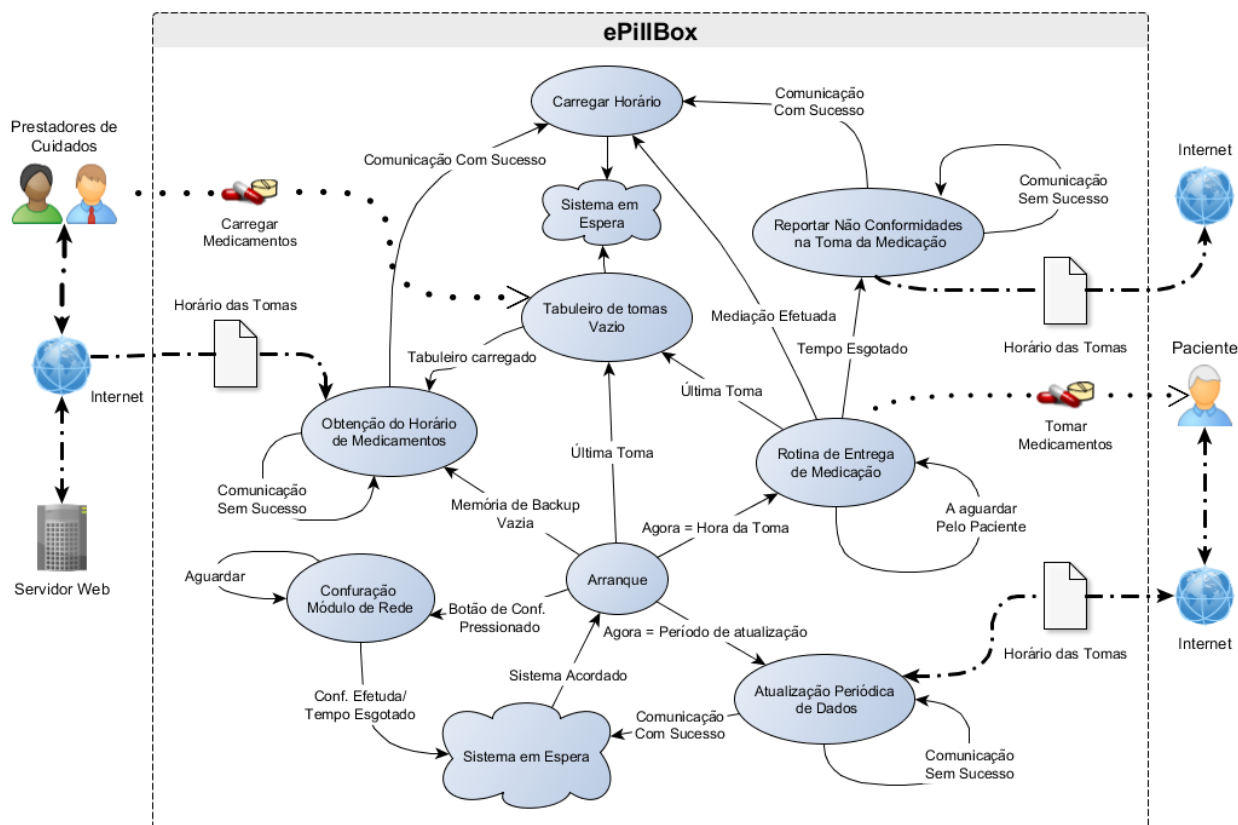


Figura 5.2– Diagrama ilustrativo dos estados do *firmware* do dispositivo.

Tal como representado na figura 5.2, o nome dos estados e as suas funções são os seguintes:

- **Arranque** – Este é o estado em que o dispositivo começa ao ser ligado à alimentação, ou quando retorna do estado de hibernação mais profundo. Nesta fase é necessário fazer verificações para aferir:
 - Qual a razão do μC estar a arrancar (se vem de um *power-up* ou se acordou do modo de hibernação mais profundo);
 - Verificação da memória de *backup*;

Também é necessário inicializar todo o *hardware* interno e externo do sistema. Por fim, é necessário iniciar o RTOS e as tarefas, terminando assim a fase de arranque.

- **Aguarda Config. Rede** – Este estado é ativado quando o utilizador pressiona um botão acessível do exterior. Nesta situação, o dispositivo entra em modo de configuração dando

acesso às suas configurações básicas. Mediante a *interface* de comunicação disponível, as configurações são diferentes:

- **Wi-Fi** – Nesta situação o dispositivo entra em modo de configuração da seguinte forma:
 - O dispositivo passa a funcionar como *Access Point* (Ponto de Acesso). O utilizador deve então utilizar um dispositivo compatível com *Wi-Fi* e, através do *browser* realizar a configuração;
 - Após entrar em modo de configuração aparecerá uma rede *Wi-Fi* com o nome de “ePillBox-AP”. O utilizador deverá ligar-se a ela;
 - Depois da ligação bem-sucedida o utilizador deverá abrir o *browser* e escrever na barra de endereço “http://1.2.3.4” e, se tudo estiver bem, aparecerá uma página para realizar a respetiva configuração;
 - Após o utilizador configurar o *SSID* da rede a utilizar e, se necessário, indicar a respetiva palavra-passe, deve carregar em guardar para terminar com sucesso o procedimento;
- **Ethernet** – Este caso é muito semelhante ao do *Wi-Fi*. O utilizador deve ligar o dispositivo ao seu computador através de um cabo de rede antes de carregar no botão de configuração.
 - O utilizador deve pressionar o botão de configuração do dispositivo para este entrar em modo de configuração;
 - Se tudo correr bem o computador indicará que foi corretamente configurado e possui IP válido;
 - O utilizador deverá abrir o *browser* e escrever na barra de endereço “http://1.2.3.4” e, se tudo estiver bem, aparecerá uma página para realizar a respetiva configuração;
 - Aí deverá fazer as configurações necessárias para que dispositivo consiga comunicar com o servidor central;
- **GSM e Bluetooth** – Neste caso não é necessário realizar nenhuma configuração. No caso do GSM os parâmetros são configurados previamente antes do dispositivo chegar às mãos do utilizador. No caso do *Bluetooth* não será necessário configurar nada na parte do dispositivo, atendendo a que todas as configurações são feitas no dispositivo utilizado para aceder à *ePillBox*.

Este estado é iniciado pelo utilizador e finalizado:

- Por ação do utilizador depois de terminar a configuração;
- Passado algum tempo, se não houver interação do utilizador com o dispositivo, este deixa este modo e entra em modo normal de funcionamento;
- **Carregar horário** – Os horários são carregados uma tabela na memória de *backup*. Dado que a aplicação *web* garante que a próxima toma se encontra no registo seguinte, evita a necessidade de utilizar algoritmos de procura na tabela para encontrar o horário da próxima toma. Neste caso, basta avançar para o próximo registo e colocar no alarme a data e hora correspondentes;
- **Sistema em espera** – É neste modo que o dispositivo estará mais tempo, no qual estará no modo de mais baixo consumo para poupar energia. Para tal, todos os periféricos externos do μC estarão desligados. Os internos, só mesmo o RTCC está em funcionamento para que este possa “acordar” o μC no horário devido. Para além do Alarme do RTCC, alguns botões estarão também funcionais para acordar o μC ;

- **Rotina de entrega de medicamento** – Inicia-se à hora pré determinada que foi programada no servidor e tem a seguinte sequência:
 - Inicia-se com um alarme visual para que o paciente se aperceba que está na hora de tomar a medicação;
 - Passado um tempo pré-determinado inicia-se, em complemento, com o visual uma sinalização sonora que vai aumentando de volume à medida que o tempo vai passando;
 - Se o paciente se aperceber chegará junto do equipamento e pressionará o botão para libertar a toma;
 - Caso a gaveta esteja colocada no devido lugar o equipamento dispensará o medicamento;
 - Caso a gaveta não esteja no devido local ficará a aguardar que a mesma seja colocada no devido local;
 - Se não for colocada dentro do tempo disponível para que a toma seja libertada, a mesma não será libertada e esta situação será comunicada;
 - Caso esteja tudo em conformidade, a toma será libertada e será registado o sucesso da entrega, para além da data e hora em que a mesma foi libertada. Nesta situação, por omissão, não será comunicado que a toma foi realizada com sucesso;
 - Caso o paciente não pressione o botão de libertar toma, esta situação será registada e comunicada para o servidor remoto, o mais rapidamente possível;
 - Por último, o alarme do RTC vai ser configurado para acordar o μC à hora da próxima toma. Após esta tarefa estar terminada o μC fica no modo mais profundo de hibernação.
- **Rotina de comunicação com servidor** – Esta rotina poderá enviar e receber vários tipos de informação, mediante o que for requerido:
 - **Inicialmente** – Enquanto a memória de *backup* não tem a informação desejada é necessário obter a mesma do servidor. Neste caso é necessário obter a tabela de tomas, e as configurações gerais do sistema;
 - **Reportar periodicamente os dados** – Num intervalo de tempo pré-definido é necessário comunicar com o servidor para atualizar a informação neste, mas também para indicar se o dispositivo está operacional. A informação enviada será sobre as tomas efetuadas com sucesso e alguns dados relativos ao funcionamento, tais como:
 - Fonte de alimentação em uso;
 - Tensão das diversas fontes de alimentação do sistema;
 - O registo dos diversos erros de funcionamento;
 - **Reportar o não cumprimento das tomas** – Neste caso, após o não cumprimento de uma toma, será comunicado de imediato, para que se possa efetuar ações para averiguar as razões e agir em conformidade;
- **Tabuleiro vazio** – Este modo é iniciado quando a tabela das tomas está vazia, no qual o equipamento indicará que não há mais tomas disponíveis no tabuleiro e o sistema estará em modo de baixo consumo a aguardar que seja abastecido novamente;
- **Em abastecimento** – Este modo inicia-se quando o compartimento das tomas é aberto. A abertura é detetada por um fim-de-curso instalado por forma a ser detetada a abertura do compartimento das tomas. Depois de aberto é esperado que este seja preenchido conforme o que foi introduzido na página web. Após o carregamento efetuado, o utilizador, deverá pressionar o botão de toma com o compartimento das tomas aberto. Se tudo correr

corretamente o equipamento emitirá um som. Depois do utilizador fechar o compartimento o dispositivo tentará obter a nova tabela das tomas. Ao fazer este procedimento indica ao servidor que o tabuleiro de tomas foi reintroduzido no seu local devido. Depois deste procedimento haverá novo sinal. Neste caso, o processo de carregamento está terminado e o equipamento entrará em modo de baixo consumo até ter nova toma a entregar.

5.2.5 - Gestão de energia

O funcionamento da *ePillbox* foi otimizado para minimizar o seu consumo. Assim, entre tomas todos os módulos internos não necessários são desligados: o microcontrolador desliga todos os seus periféricos, com exceção do RTCC. O RTCC não pode ser desligado porque é ele que reativa o microcontrolador quando a hora da toma chega. Na figura 5.3 pode ver-se um gráfico que ilustra o exposto. A rotina da toma inicia-se quando chega a hora de efetuar a toma e termina quando uma de duas coisas acontecem:

- O utilizador efetuou a toma;
- O utilizador deixou passar o tempo máximo que tinha para efetuar a toma.

Apesar do tempo que o utente pode demorar a executar a rotina da toma ser variável, será sempre inferior ao tempo entre tomas, o que faz com que haja uma poupança significativa de energia.

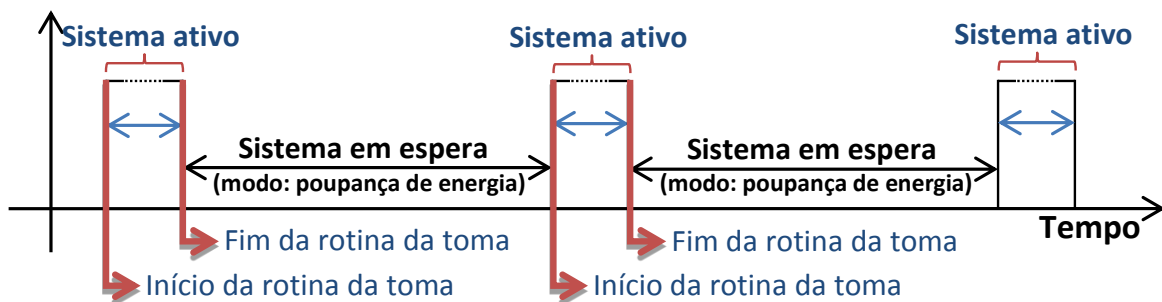


Figura 5.3 – Diagrama de temporal dos estados da *ePillBox*.

5.3 - A Interface de Programação de Aplicativos (API)

A *interface* de programação de aplicativos ou, em inglês, *Application Programming Interface*, tem como objetivo realizar a *interface* entre o servidor de base de dados e o dispositivo *ePillBox*. A troca de informação entre o equipamento e o servidor *web* é feita através do protocolo *JSON-RPC*.

5.3.1 - Comunicação *ePillBox* ↔ API

O diagrama da figura 5.4 ilustra a forma como é realizada uma troca genérica de informação entre a *ePillbox* e o servidor remoto. O servidor utilizado é um servidor WEB com a linguagem PHP e uma base de dados em *MySQL*. O protocolo utilizado é o *JSON-RPC* sobre *HTTP*.

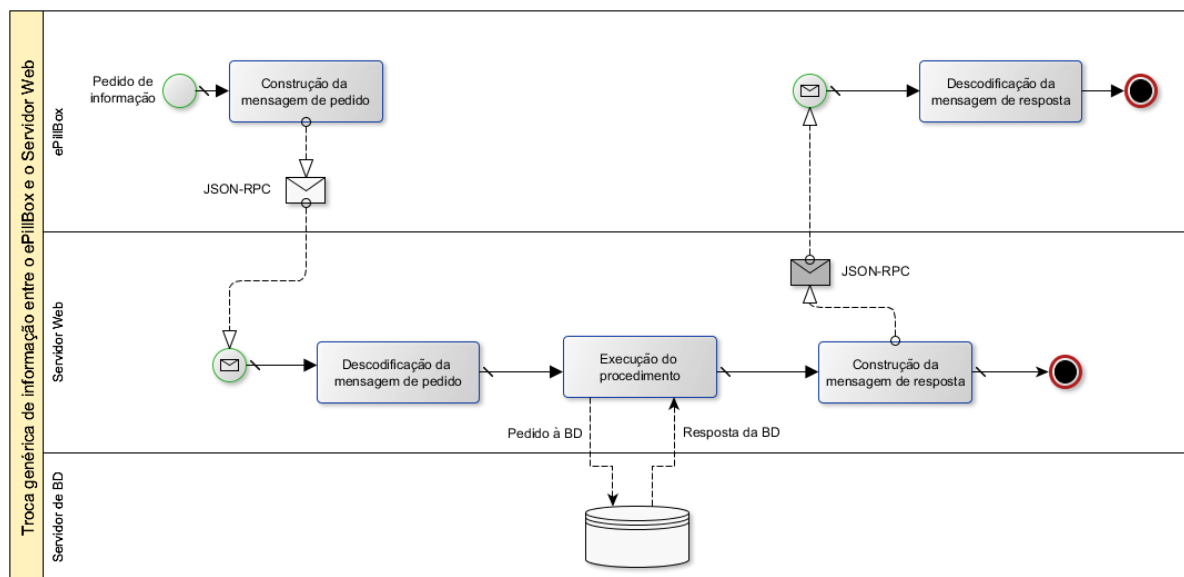


Figura 5.4 – Exemplo de uma troca de mensagens entre a ePillBox e o servidor Web.

5.3.2 - O protocolo JSON-RPC

O protocolo utilizado para realizar a troca de informação entre os dispositivos foi o *JSON-RPC – versão 2.0*. O *JSON-RPC* é um protocolo simples de chamada de procedimento remoto codificado em *JSON* (RFC 4627), que permite a comunicação bidirecional entre o servidor e o cliente e onde cada uma das pontas pode chamar a outra. Também permite o envio de múltiplas chamadas de uma ponta a outra que podem ser respondidas fora de ordem [65]. Este protocolo não define a camada de transporte e pode ser utilizado sobre o protocolo HTTP ou diretamente através de *sockets* TCP/IP.

5.3.3 - Formato do objeto de pedido

O formato do objeto de pedido definido pelo “JSON-RPC – versão 2.0” deve ser um único objeto JSON e conter os seguintes membros:

- **jsonrpc** – Tem que conter uma *string* que identifica a versão do protocolo a utilizar; neste caso tem que ser exatamente – “2.0”;
- **method** – Deve conter uma *string* como o nome do método (função) a ser chamado. Os nomes iniciados pela palavra “rpc” seguida de um ponto estão reservados para uso interno do protocolo;
- **params** – Deve conter os valores dos parâmetros a serem utilizados na execução do método. O formato a utilizar será em matriz ou um objeto JSON. Este parâmetro pode ser omissos;
- **id** – Trata-se de um identificador que é atribuído pelo cliente; este *id* pode ser um número ou uma *string*;

Todos os membros são obrigatórios com exceção do “*params*”.

5.3.3.1 - Formato do objeto de notificação

O objeto de notificação é um objeto de pedido, mas retirando o membro *id*. Este objeto pode ser utilizado quando o cliente não necessita de resposta por parte do servidor. Como o servidor não responde à notificação não é possível assegurar que esta tenha sido corretamente tratada pelo servidor.

5.3.4 - Formato do objeto de resposta

O formato do objeto de resposta definida pelo “JSON-RPC – versão 2.0” deve ser somente um objeto JSON contendo os seguintes membros:

- **jsonrpc** – Tem que conter uma *string* que identifica a versão do protocolo a utilizar; neste caso tem que ser exatamente – “2.0”;
- **result** – Contém o resultado da execução do método previamente chamado e a sua presença na trama depende do sucesso do pedido:
 - É obrigatório caso o procedimento tenha sido executado com sucesso;
 - Não pode existir caso o procedimento tenha falhado;
- **error** – Membro deve ter o formato definido no ponto 5.3.4.1 - e:
 - Não pode existir caso o procedimento tenha sido executado com sucesso;
 - É obrigatório caso o procedimento tenha falhado;
- **id** – Este membro é obrigatório e terá os seguintes valores:
 - Caso o pedido tenha sido executado com sucesso deverá ser igual ao *id* do pedido;
 - Caso haja um erro na detecção do *id* no pedido, tal como, por exemplo, um erro de interpretação, ou um pedido inválido, o seu valor deverá ser nulo (ver ponto 4 da tabela 5.1).

5.3.4.1 - Formato do objeto “error”

Quando num pedido de RPC ocorre um erro, a trama de resposta tem de conter um membro “error”. Esse membro é um objeto com os seguintes membros:

- **code** – É constituído por um número inteiro que indica o tipo de erro ocorrido;
- **message** – É uma *string* com uma breve descrição do erro;
- **data** – É um membro que contém mais informação acerca do erro, o qual pode ser omissa.

Mais informação acerca do protocolo JSON-RPC poderá ser encontrado em [65]. Na tabela 5.1 estão disponíveis exemplos de alguns tipos de objetos utilizados no protocolo JSON-RPC.

Tabela 5.1 – Lista de alguns tipos de objetos utilizados no protocolo JSON-RPC.

1.a)	→ { "jsonrpc": "2.0", "method": "somar", "params": [42, 23], "id": 1 }
1.b)	← { "jsonrpc": "2.0", "result": 65, "id": 1 }
2.a)	→ { "jsonrpc": "2.0", "method": "atualizar", "params": [1,2,3,4,5] }
2.b)	→ { "jsonrpc": "2.0", "method": "a" }
3.a)	→ { "jsonrpc": "2.0", "method": "foobar", "id": 10 }
3.b)	← { "jsonrpc": "2.0", "error": { "code": -32601, "message": "Procedure not found." }, "id": 10 }
4.a)	→ { "jsonrpc": "2.0", "method": "foobar", "params": "bar", "baz" }
4.b)	← { "jsonrpc": "2.0", "error": { "code": -32700, "message": "Parse error" }, "id": null }
Legenda: 1) Objeto de pedido (a), Objeto de resposta (b); 2) Objeto de notificação com parâmetros(a) e sem parâmetros (b); 3) Objeto de pedido que chama um método que não existe (a), objeto de resposta com o erro e respetiva descrição (b); 4) Objeto de pedido inválido(a), Objeto de resposta ao pedido inválido (b).	

5.3.5 - Listagem de procedimentos da API

Os procedimentos disponíveis na API do servidor web são os mínimos para permitir a troca de informação entre o dispositivo e o servidor. Os procedimentos implementados são os seguintes:

- **Set_deviceRegistration** – Regista o dispositivo e as suas características na base de dados (BD);
- **update_deviceRegistration** – Atualiza o registo do dispositivo e as suas características na BD;
- **get_doseTable** – Pede ao servidor a nova tabela de medicamentos;
- **update_doseTable** – Atualiza a tabela das tomas com as tomas já efetuadas;
- **get_generalConfig** – Pede ao servidor as configurações gerais para o dispositivo;
- **get_dateTime** – Pede a data e hora do servidor;
- **report_PanicEvent** – Reporta ao servidor que o botão de pânico foi pressionado;

No anexo D estão presentes mais detalhes sobre as respetivas funções.

5.4 - Sistema de Informação Web

Neste ponto descreve-se o desenvolvimento *web* para o sistema de informação *ePillbox*. Na Seção 5.4.1- é apresentada uma breve introdução sobre os aspetos relevantes do desenvolvimento do sistema de informação (web). Na seção 5.4.2- são apresentadas as principais escolhas sobre a documentação do projeto, as *frameworks* selecionadas, ferramentas, bibliotecas, linguagens de programação e banco de dados. Por fim, a seção 5.4.3 - documenta o sistema de informação.

5.4.1 - Introdução

Desde sempre, os sistemas de informação colocaram muitos desafios os seus criadores. Muitas ferramentas, como linguagens de modelização, gráfica ou textual, existem com o objetivo de cobrir os aspetos, desde as especificações de montagem até à fase de codificação, a fim de gerar um protótipo para mostrar ao utilizador final. Apesar das excelentes especificações de montagem, geralmente são necessárias algumas mudanças – estes são alguns dos aspetos da manutenção associada ao desenvolvimento de sistemas de informação. Para resolver a este problema foram desenvolvidas algumas linguagens orientadas para a sua solução. Dado que o desenvolvimento de *software* é uma tarefa muito específica, dependente do setor da tecnologia e atividade em questão, a solução para o problema da manutenção de *software* não é uma tarefa fácil.

Com o objetivo de criar modelos visuais para sistemas de *software* orientados a objetos foi criado o UML (*Unified Modeling Language*) que é padronizada pela ISO/IEC 19501:2005. No entanto, muitos autores argumentam sobre a sua inconsistência, porque contém muitos diagramas e construções que são redundantes ou raramente utilizadas [66] e, por causa do UML ser extremamente complexo [67], quando comparado com outras ferramentas, como a linguagem DOT [68] ou GraphML [69]. Apesar das críticas, a UML continua a ser muito utilizada na modelação de sistemas de informação.

Hoje em dia há muitas aplicações que são desenvolvidas em ambiente *web*, normalmente ajudadas por telas com formulários massivamente interativos (páginas web). Outro padrão que é agora aceite para preencher esta lacuna (UML perde claramente quando se torna necessário criar *interfaces* de utilizador complexas e interações) é o IFML (*Interaction Flow Modeling Language*), que é dedicado à definição da dinâmica de “interação com o utilizador” entre a aplicação e o utilizador. Desde março de 2013, IPMF é agora um padrão gerido pelo OMG (*Object Management Group*), [70]. Outra norma importante é a BPMN (*Business Process Model and Notation*) [70], que possibilita a compreensão dos procedimentos internos do “negócio” e permite as transações entre instituições, através de uma notação gráfica.

Esses padrões são suportados por algumas ferramentas de *software* para implementar a documentação dos sistemas de informação e as suas especificações. Para UML, RationalRose,

Enterprise Architect e Papyrus, TopCased, Modelio, WhiteStarUML [71] são ferramentas gratuitas que são capazes de gerar código C++, C# e Java; para o IFML, o *WebRatio* [72] é uma ferramenta comercial com 10 anos de experiência no desenvolvimento do padrão IFML. Muitas outras ferramentas podem ser encontradas na *web* (por exemplo, para UML: Umbrello, Poseidon, MagicDraw UML, ArgoUML, Oracle JDeveloper). Os dados podem ser transferidos de ferramenta para ferramenta utilizando o formato XML.

No desenvolvimento de aplicações *web* alguns aspetos devem ser considerados, como é o caso da linguagem de implementação (C, C++, PHP, Java, Ruby, C#, F#, D, JavaScript (usando Node.js)) [73]. As *frameworks* que devem ser usadas são outras das questões que se devem ter em consideração. Outro problema é a separação da apresentação gráfica da lógica da aplicação, pois em programação *web* o *design* gráfico é uma questão importante. Também deve ser considerado se a separação é feita no servidor ou no navegador (cliente). O MVC (*Model View Controller*) é uma arquitetura de *software* que separa a lógica da aplicação da interação com o utilizador; este foi inventado em 1970 para o *Smalltalk*, mas existem algumas outras variantes. Para o servidor *Phalcon* [74] e para o cliente o *JavaScriptMVC* são uma opção interessante; *Backbone* [75] inclui também a filosofia MVC com o *REST* (*Representational State Transfer*) que implementa os serviços *web* (só para citar alguns).

Outro aspeto importante tem a ver com a forma como a comunicação entre cliente e servidor é realizada. Deve-se questionar se devemos usar serviços *web*, ajax, refrescamento normal de página, ou outra tecnologia. Para os serviços *web* algumas normas são igualmente importantes, tais como a WSDL (*Web Service Description Language*) [76]. A WADL (*Web Application Description Language*) [28] é uma maneira de especificar *interfaces* de dados usando métodos de serviços *web* REST. O WSDL, após a revisão 2.0, também é capaz de usar, por completo, os métodos REST. Para o lado do cliente estão disponíveis algumas linguagens, tais como o HTML5, o CSS3, e JavaScript; os applets (programação java), o *Silverlight* (Microsoft), *Adobe Flash / Flex* [77] são opções para usar em gráficos, animações, reprodução de vídeo de alta definição ou formulários interativos. O *Dart* [77] é uma outra linguagem de programação emergente proposta pela Google para substituir JavaScript; no entanto, ele não é aceite pelos grandes fabricantes de tecnologias *web*, tais como navegadores *web* ou sistemas operativos. O *Chrome* (navegador da Google) suporta nativamente esta nova linguagem, incorporando uma máquina virtual *Dart*. Os navegadores que não suportam (cujos fabricantes haviam dito que nunca o suportariam de forma nativa) o Google desenvolveu uma biblioteca JavaScript que, primeiro compila o aplicativo para JavaScript e, em seguida, executa-o no navegador padrão do sistema.

Estão disponíveis muitas *frameworks*, quer direcionadas para a programação do servidor ou para a programação do cliente mais servidor, tais como o *Opa*, *Terus*, *CoffeeScript*, *Pyjamas*, *TypeScript*, etc [78]. *Opa*, *Pyjamas*, *TypeScript* e *CoffeeScript* são novas linguagens que compilam linguagens de alto nível, tais como JavaScript, enquanto *Terus* é uma ferramenta gráfica que, após a compilação, gera os códigos de cliente e servidor.

Os Sistemas de gestão de conteúdos, como *Joomla*, *Drupal* e outros, também são opções importantes devido a muitos pacotes de *software* que já existem para ele, permitindo reduzir o tempo de desenvolvimento até a aplicação entrar no mercado (*time-to-market*).

As Bases de dados também são importantes, devido à maneira de interagir com elas. Uma questão em aplicações de bases de dados são as mudanças na estrutura das tabelas, que se propagam para a lógica da aplicação e *interfaces* gráficas. Esta situação consome muitas horas de programação, e pode implicar a introdução de erros, a propagação de inúmeras mudanças e a necessidade de alterar comandos SQL. Para eliminar a dependência de comandos SQL existem vários pacotes, como o *LINQ* [79], *ORM*, *XQuery* e outros; no entanto, alguns autores argumentam que essa é uma diferença de impedância objeto-relacional [80].

Dependendo da linguagem de programação, estão disponíveis boas *frameworks* [74], [81], [82] (só para citar alguns), tais como as seguintes:

- Python – Pilares, TurboGears, Zope, Django;

- PHP – Yii, sinfonia, Kohana, PPI, Phalcon (extensão C para PHP), FuelPHP, ScriptCase ;
- Ruby – Ruby on Rails;
- Groovy – Grails;
- Frameworks para o navegador do cliente – html5boilerplate, Twitter Bootstrap.

Relativamente às linguagem de *webpages* – HTML – para manipulação DOM existem bibliotecas muito populares, como o *jQuery*, *Mootools*, *Prototype*, *Dojo*, *YUI* [83] que permitem páginas interativas no cliente. O *jQuery* é uma biblioteca muito popular, com muitos *plugins* para a conceção da *interface* gráfica e interativo. O *ExtJS* (comercial) é outro exemplo de uma boa biblioteca.

AlloyUI é um exemplo de uma biblioteca *JavaScript* que permite a criação de *templates*, carregamento modular, *MVC*, manipulação *DOM* e componentes da *interface* com o utilizador no navegador [83].

O *software* de Controlo de Qualidade é outro aspeto importante no desenvolvimento de sistemas de informação.

Todas estas perguntas e opções são decisões que a equipa de desenvolvimento deve responder e não são fáceis devido às inúmeras *frameworks* e ferramentas atualmente disponíveis.

5.4.2 - Ferramentas de desenvolvimento de sistema de informação e *Pillbox*

Para desenvolver o sistema de informação *web* foram testadas algumas ferramentas para escolher as mais adequadas:

- Para o servidor foi selecionada a *framework* *Phalcon* 1.2.3; esta escolha permite usar o *PHP* como linguagem de servidor com a possibilidade de usar *MVC*, *ORM*, e *templates* HTML (Volt), etc. Este *framework* é uma extensão do *PHP* codificado em C, o que permite alta performance em serviço web em conjunto com o servidor web Apache.
- No lado do cliente foi selecionada a biblioteca *AlloyUI*, que é baseada em *YUI3* e, como já indicado, suporta *templates*, carregamento modular, *MVC*, componentes de manipulação de *DOM* e *interface* de utilizador. A linguagem de trabalho no navegador é a *JavaScript*.

Para documentar o projeto, foi selecionada UML baseada na ferramenta *open source* *WhiteStarUML*.

Porque a linguagem *PHP* é interpretada, se o desempenho ou o tempo de carregamento do *site* for um problema, há muitas hipóteses para compilar os programas escritos nesta língua. As hipóteses são as seguintes [84]:

- ***bcompiler*** – É uma extensão *PECL*, ainda numa fase experimental (programas compilados podem não correr exatamente como na versão interpretada);
- ***hhvm*** – É *JIT* (Just-in-Time), um compilador desenvolvido pelo engenheiros do *Facebook*;
- ***APS Compiler*** – *APS* é um compilador de código aberto para *PHP* com suporte para *plugins*, tendo estado em desenvolvimento até 2011;
- ***Phalanger*** – É um interpretador completo de *PHP* e compilador para a *framework* *.NET/Mono* desenvolvidos em *C#*.

Se necessário, este projeto pode usar a opção "bcompiler", porque há uma maior garantia de compatibilidade com as novas versões do *PHP*.

Existem outras opções que podem ser tidas em consideração, mas estas ferramentas possuem licenças MIT ou BSD, o que permite dar uma direção comercial para o projeto (caso contrário, este projeto, não passará de um mero exercício académico).

5.4.3 - Informações sobre Web Sistema e *Pillbox*

Os requisitos para este sistema são os seguintes:

- Um médico irá prescrever medicamentos a um paciente; em princípio, a doença é crónica, levando a que o medicamento tenha que ser tomado regularmente e em horários fixos (semanas ou meses,...) – com dosagem e cronograma;
- A prescrição pode ter um ou vários medicamentos, com diferentes períodos de tempo entre as várias tomas;
- Os cuidadores são responsáveis por agendar o medicamento num calendário semanal de acordo com o médico e a prescrição de medicamentos – também programam e ativam os alarmes fornecidos pelo *firmware ePillBox*;
- Os cuidadores podem adicionar novos cuidadores ou transferir o cuidado para outro;
- Um cuidador pode cuidar de vários pacientes (*ePillBoxes*);
- O *ePillBox*, na presente versão tem 35 compartimentos que podem ser usados numa base de tempo semanal, com cinco períodos de dosagem por dia – o cuidador é quem programa o calendário da medicação – os planos futuros preveem o uso de diferentes *ePillBoxes* com diferente número de compartimentos, dependendo do estado de saúde do paciente;
- Cada *ePillBox* pode ser identificado por um número de série (identificador único) determinado pelo identificador de processador – comunicação é feita por TCP/IP.

Mysql é o principal motor base de dados para registo da informação; o esquema das tabelas está ilustrado na figura 5.5. O *site* permitirá três diferentes níveis de acesso – administração; cuidadores; e pacientes ou familiares do paciente:

- **Administração** – O utilizador insere a informação sobre as *ePillBox* (números de série);
- **Cuidadores** – Após o registo no *site*, estes irão incluir a informação do paciente e o número de série da *ePillBox*, por forma a relacioná-los. Após esta etapa podem preencher o calendário da medicação numa base regular para todos os pacientes pelos quais são responsáveis;
- **Os pacientes ou familiares do paciente** – Estes podem verificar as informações relacionadas com o calendário da medicação, alarmes, medicamentos tomados e não tomados, alguma situação urgente, e assim por diante, e decidir quais os alarmes ativos que deseja receber.

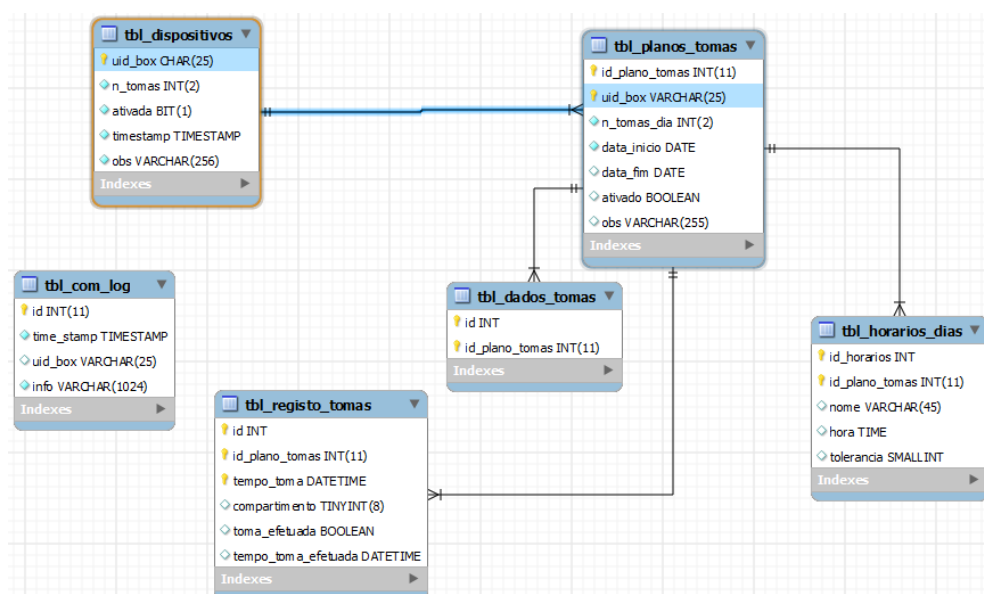


Figura 5.5 – Diagrama das tabelas Mysql usando a ferramenta MySQL Workbench – Tabelas normalizada na terceira forma normal;

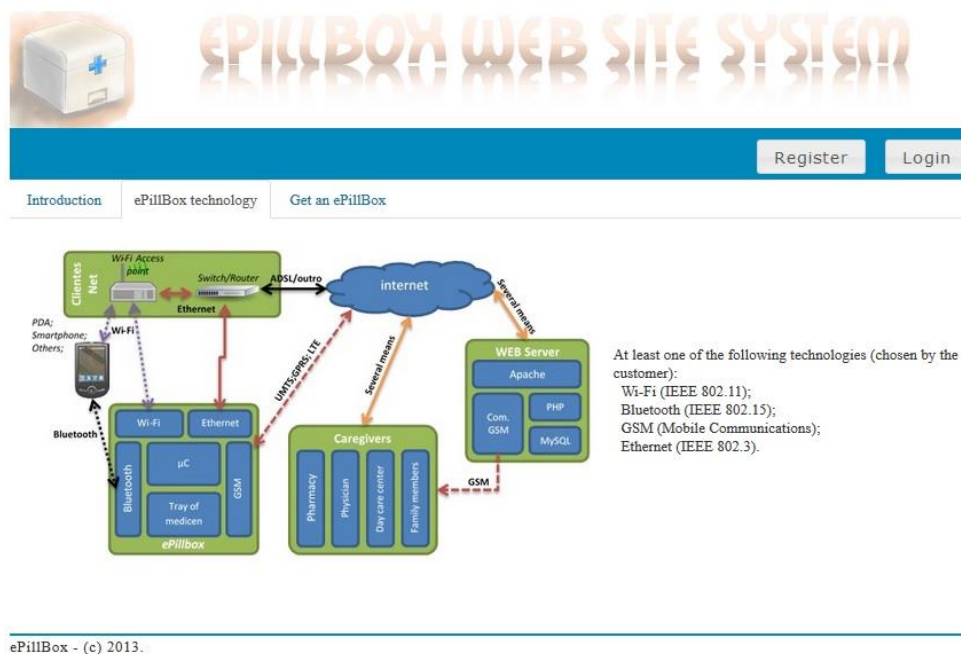


Figura 5.6 – ePillBox Sistema Web Site – frontend principal.

The screenshot shows the 'PERSONAL INFORMATION' registration form on the ePillBox Web Site System. The form includes the following fields and instructions:

- Username:** Please enter your full name.
- Password:** Please enter a password.
- Password (*):** Confirm your password.
- Email:** Please enter our email.
- Confirm Email:** Confirm our email.
- Secu. Question:** Security Question
- Secu. Answer:** Security Answer

At the bottom right, there are two buttons: 'Típico' and 'CREATE USER'. A progress bar at the top indicates 'Step 1/4: 25%'.

ePillBox - (c) 2013.

Figura 5.7 – ePillBox Sistema Web Site – inscrições novo utilizador;

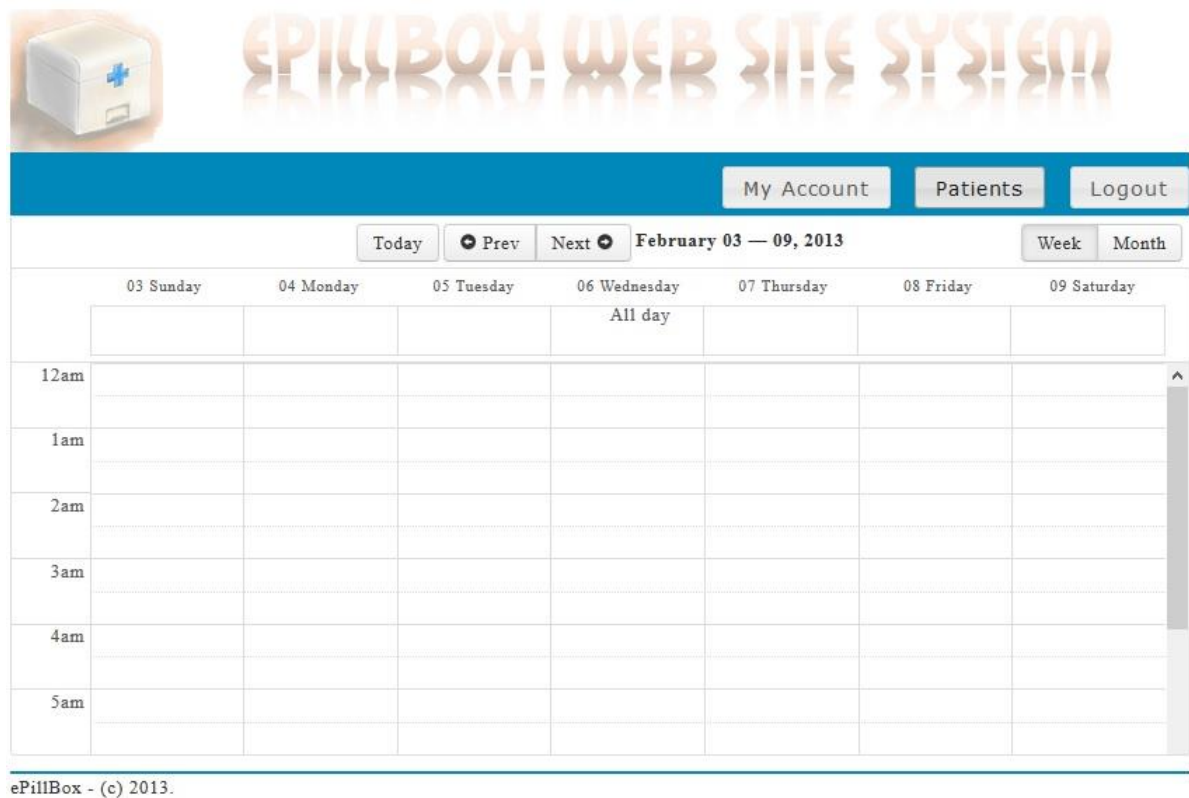


Figura 5.8 – ePillBox Sistema Web Site – Medicação Schedule.

5.5 - Sumário

O neste capítulo foi abordado o desenvolvimento de *software* levado a efeito no projeto. Foi feita a descrição do ambiente de desenvolvimento, apresentada a forma como foi efetuada a definição das secções de código, as funcionalidades implementadas e a análise detalhada de cada um dos blocos.

Capítulo 6 - Validação do dispositivo

Este capítulo apresenta os testes que foram realizados para assegurar que o *software* e o *hardware* produzidos, neste projeto, têm o funcionamento e desempenho de acordo com os requisitos inicialmente definidos.

6.1 - Testes efetuados ao *hardware*

Para verificar o correto funcionamento do *hardware* foi realizado um conjunto de testes. Dado que alguns dos componentes utilizados, nomeadamente o μC e o Conversor DC/DC, não possuem encapsulamentos “amigáveis” para utilização em placas de ensaios (*breadboards*), foi decidido mandar fazer as PCI e, posteriormente, realizar os testes, atendendo a que os circuitos utilizados já davam alguma garantia relativa ao seu funcionamento, devido a terem sido elaborados com base nos esquemas propostos pelo fabricante, como é o caso do conversor DC/DC. No caso do μC , o *design* dos circuitos seguiu as recomendações do fabricante, tendo sido alguns testes executados na placa de desenvolvimento, *STM32F4 – Discovery*. Ainda relativamente ao processador, existem suportes que permitem a colocação e remoção do processador [85]; no entanto, o custo desta solução é significativamente superior ao da produção das PCI [86]. Os módulos produzidos em PCI foram separados para evitar que um erro num módulo inviabilize o funcionamento de outro.

Inicialmente foram produzidos unicamente três módulos:

- Placa do μC ;
- Fonte de alimentação;
- Controlo dos atuadores das tampas;

6.1.1 - Produção das PCI

Inicialmente estava previsto produzir as PCI através de CNC recorrendo à capacidade interna do grupo de trabalho. No entanto, a determinada altura abandonou-se a ideia já que o equipamento disponível não tinha a resolução necessária para produzir as PCI. A alternativa foi recorrer a serviços externos de produção de PCI, tendo o fornecedor escolhido capacidade de produção de pequenas quantidades de PCI a um preço acessível [86]. As PCI produzidas podem ser vistas na figura 6.1.

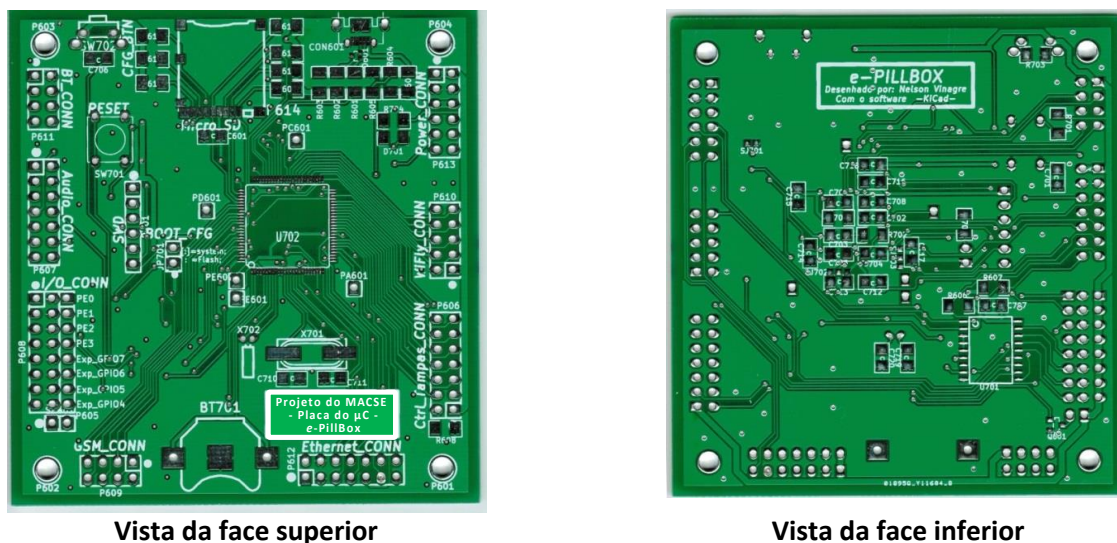


Figura 6.1– Fotografia das PCI do módulo do μC .

6.1.2 - Soldadura dos componentes

Os componentes foram soldados pelo processo manual, com recurso a uma estação de soldadura com ferro de soldar e ventilador de ar quente. O processo de soldadura é um pouco repetitivo e sujeito a erros, nomeadamente à troca de componentes ou, no caso dos componentes que têm polaridade, à troca de polaridade. Na figura 6.2 pode ver-se o aspeto da PCI do módulo do microcontrolador com e sem os conetores.



Vista de topo – Sem conetores

Vista de topo – Com conetores

Vista de baixo – Com conetores

Figura 6.2 – Fotografia da PCI do módulo do μ C parcialmente soldada (imagem da esquerda) e completamente soldadas (Imagens do centro e direita).

A figura 6.3 mostra a PCI com os módulos de controlo das tampas e da fonte de alimentação juntas. Os módulos estão separados; no entanto, por conveniência de fabrico, fizeram-se as duas placas juntas. Depois de soldar torna-se necessário separá-las, conforme se indica através da linha a tracejada. Após este procedimento fica-se com duas placas independentes.

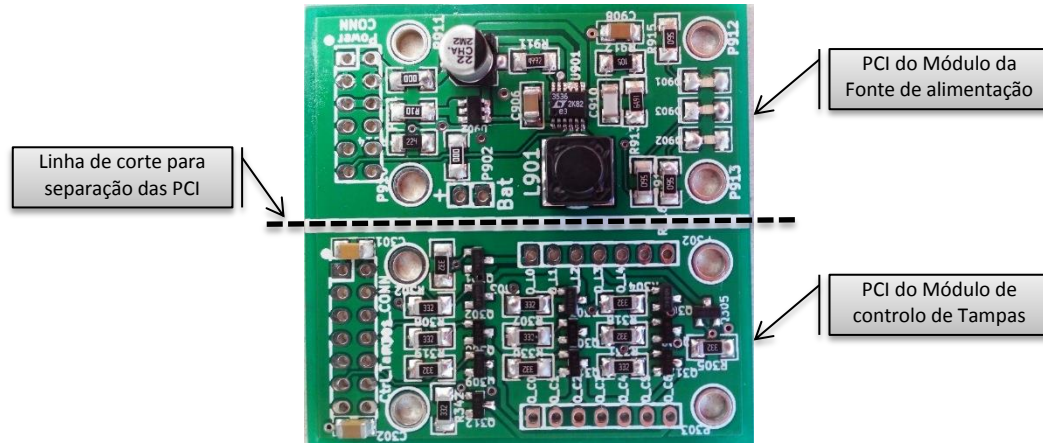


Figura 6.3 – Fotografia da PCI dos módulos da fonte de controlo das tampas e da fonte de alimentação.

Por fim inspecionaram-se alguns dos componentes com as soldaduras mais minuciosas com uma lupa para aferir a qualidade das soldaduras. Na figura 6.4 podem-se algumas dessas soldaduras em pormenor.

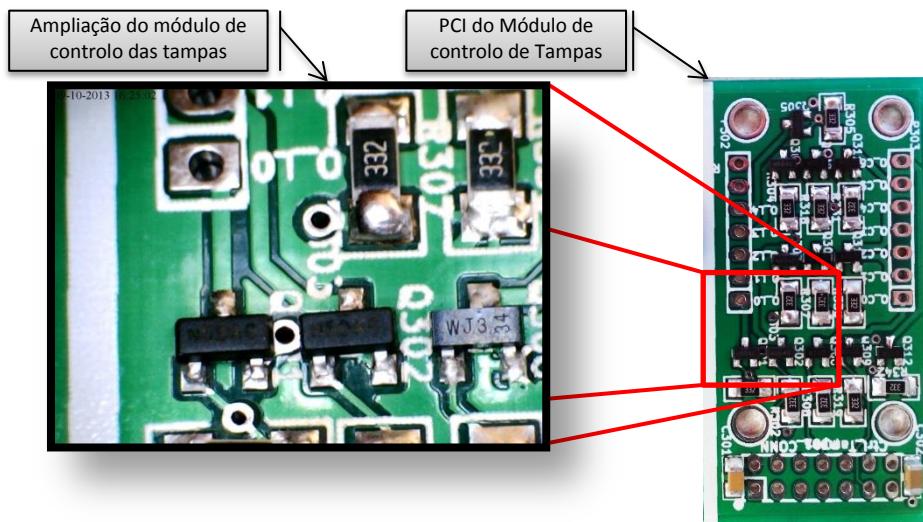


Figura 6.4 – Ampliação de algumas soldaduras para verificação da qualidade das mesmas.

6.1.3 - O módulo da fonte de alimentação

Para realizar os testes do módulo de alimentação foi utilizada uma fonte de alimentação de laboratório para fornecer uma tensão à entrada e, para verificar o valor de saída, foi utilizado um osciloscópio. Se tudo estivesse a funcionar conforme o pretendido o circuito deveria colocar na sua saída uma tensão de 3,3V mediante a colocação de uma tensão na entrada de 1,8V a 5,5V.

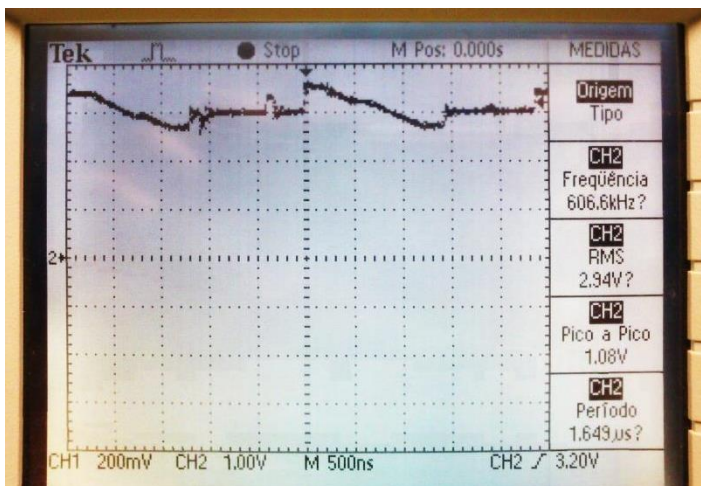


Figura 6.5 – Forma de onda obtida na saída da fonte de alimentação.

Dados recolhidos:

- $V_{RMS} = 2,94V$
- $V_{ripple} = 1,08V$

Pelos valores obtidos facilmente se conclui que o circuito não está a funcionar. Neste caso, a tensão eficaz está abaixo do esperado (3,3V), mas o mais problemático é a tensão de *ripple*, que não é admissível para o circuito em causa, como se pode ver na figura 6.5.

Foram tentadas algumas alterações em alguns componentes do circuito para averiguar qual o problema, mas não foi possível chegar a nenhuma conclusão.

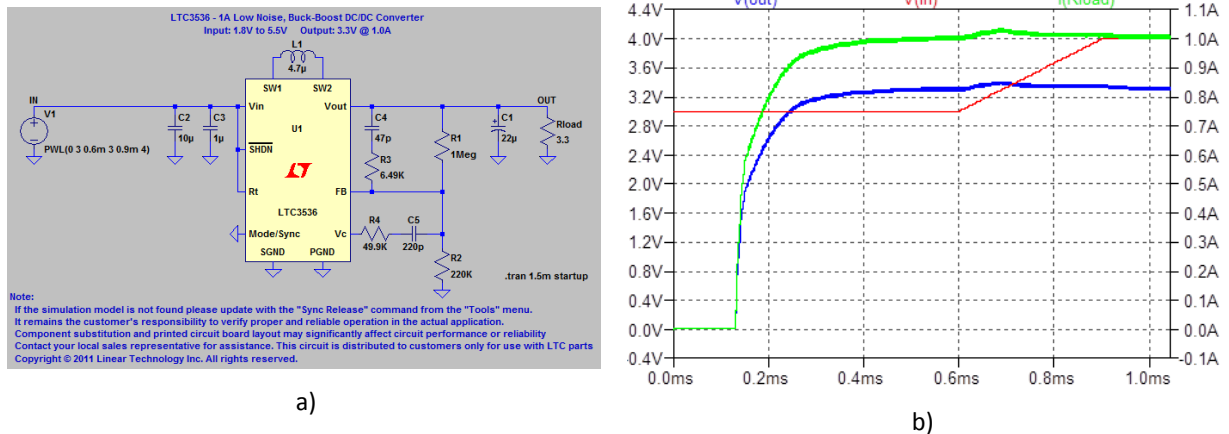


Figura 6.6 – Circuito simulado a) e resultado da simulação b) no software LT-Spice IV.

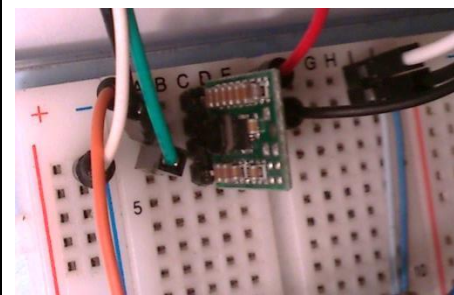
Utilizando o *software* de simulação 'LTspice IV' foi possível simular o funcionamento do circuito (figura 6.6 a) e obter as formas de onda esperadas na saída da fonte de alimentação (figura 6.6 b). Como se pode ver na figura 6.6 a saída do conversor DC-DC ($V_{(out)}$) mantém-se entre 3,2V e 3,3V depois de um período de arranque de cerca de 300 μ s. Isto era o comportamento esperado do circuito da PCI.

6.1.4 - Alternativa ao módulo da fonte de alimentação

Dado que a placa desenvolvida não funcionava convenientemente foi necessário procurar uma nova solução. A solução encontrada foi o módulo S7V8F3 da *Pololu* [87] que se pode ver na figura 6.7a). Segundo o fabricante, o módulo consegue fornecer uma corrente máxima de 1A, possui uma gama de tensões de entrada que vai desde os 2,7V aos 11,8V, uma eficiência máxima de 90% e possui uma corrente quiescente de 100 μ A.



a²²⁾



b)

Figura 6.7 – Imagem do módulo S7V8F3 a) e imagem do módulo montado numa placa de ensaio b).

O módulo foi testado utilizando uma fonte de alimentação de laboratório e diversas cargas com vários consumos. Variando a tensão de entrada e a corrente de saída, dentro dos parâmetros do módulo, verificou-se que mantinha uma tensão fixa na sua saída de 3,3V, como o esperado.

6.2 - Testes efetuados ao Firmware

6.2.1 - Teste do módulo do μ C

O teste do módulo iniciou-se com a verificação da operacionalidade dos CI e das respetivas ligações. Relativamente ao μ C o primeiro teste consistiu em lhe ligar o sistema de *debug* e verificar se ele respondia normalmente. Este teste foi bem-sucedido e o próximo passo foi verificar se todos os pinos

²² Fonte: <http://a.pololu-files.com/picture/0J4482.600.jpg?301617c21df66c0975401399d76b026a>

estavam corretamente soldados. Para tal colocou-se uma tarefa a correr no μC que colocava todos os pinos de um porto a gerarem uma onda quadrada de 10Hz e com ajuda do osciloscópio, verificou-se se o sinal chegava ao local apropriado. Utilizando esta metodologia possibilitou-se a identificação dos pinos do μC mal soldados para posteriormente serem retocados.

Relativamente ao chip de expansão de portas (U701) foi realizado um teste semelhante para testar os seus pinos de entrada/saídas.

6.2.2 - Teste do módulo de controlo dos atuadores das tampas

Para realizar os testes deste módulo foi necessário realizar um circuito que imitasse as ligações dos atuadores nos tabuleiros das tomas. Então utilizou-se uma placa de ensaio para montar um circuito matricial onde se utilizam LED em conjunto com resistências para substituir os atuadores. Os LED permitem uma fácil conformação visual de que o atuador está ativo e a resistência, que possui um valor semelhante ao que o atuador apresenta, permite simular o consumo. O circuito referido pode ser visto na figura 6.8.

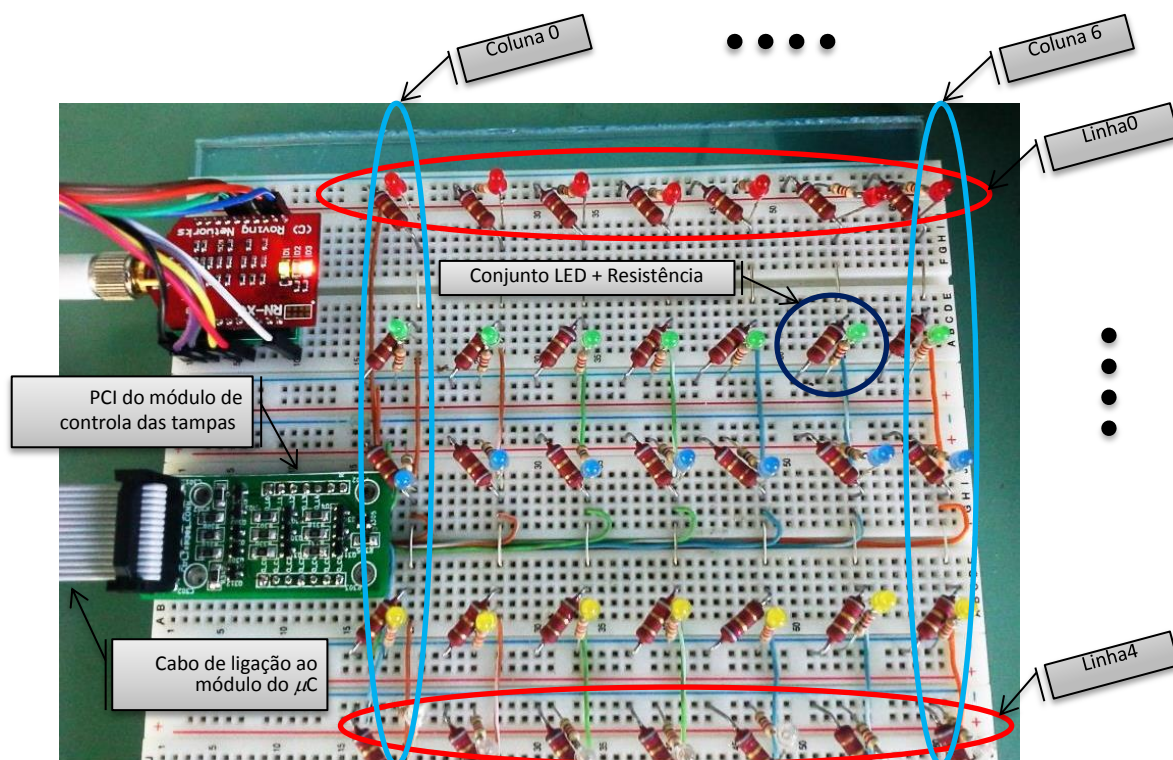


Figura 6.8 – Matriz de LED para simulação dos atuadores das tampas.

A verificação do seu funcionamento foi feito com base num programa que atuava as cargas sequencialmente e o seu funcionamento pode ser confirmado visualmente através do acendimento do respetivo LED, como se pode ver na figura 6.9.

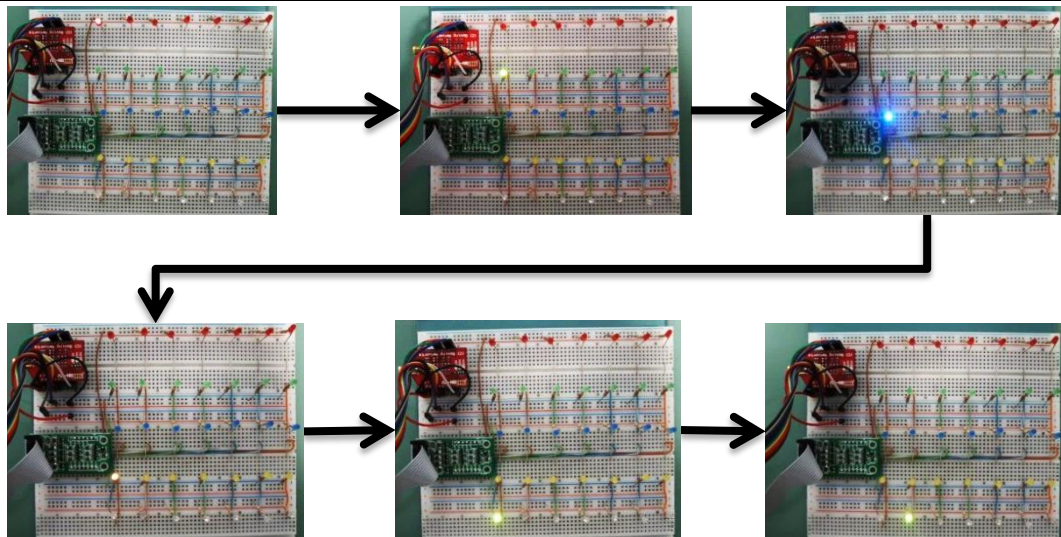


Figura 6.9 – Sequência de imagens que ilustram a atuação das tampas.

6.2.3 - Teste dos módulos de comunicação

Dos quatro módulos de comunicações só o GSM é que não foi testado. Os restantes foram testados, primeiro no *kit* de desenvolvimento (*STM32F4 – Discovery*) de forma individual e somente o módulo de *Wi-Fi* é que foi testado em conjunto com as placas desenvolvidas.

6.2.3.1 - Teste do módulo Bluetooth

O módulo *Bluetooth* foi sempre utilizado para assegurar a ligação de uma UART do μC ao PC ou outro dispositivo com Bluetooth para realização do *debug* do programa à medida que este ia sendo desenvolvido. Assim, utilizando o programa “Bluetooth Terminal” e no μC um interpretador de comandos, foi possível testar a comunicação entre eles. Na figura 6.10 pode ver-se o módulo *Bluetooth* ligado à placa do μC e uma captura de ecrã do programa “Bluetooth Terminal”²³ a mostrar os comandos enviados e as respostas devolvidas pelo μC . Dado que há troca de informação, o módulo funciona conforme o esperado.

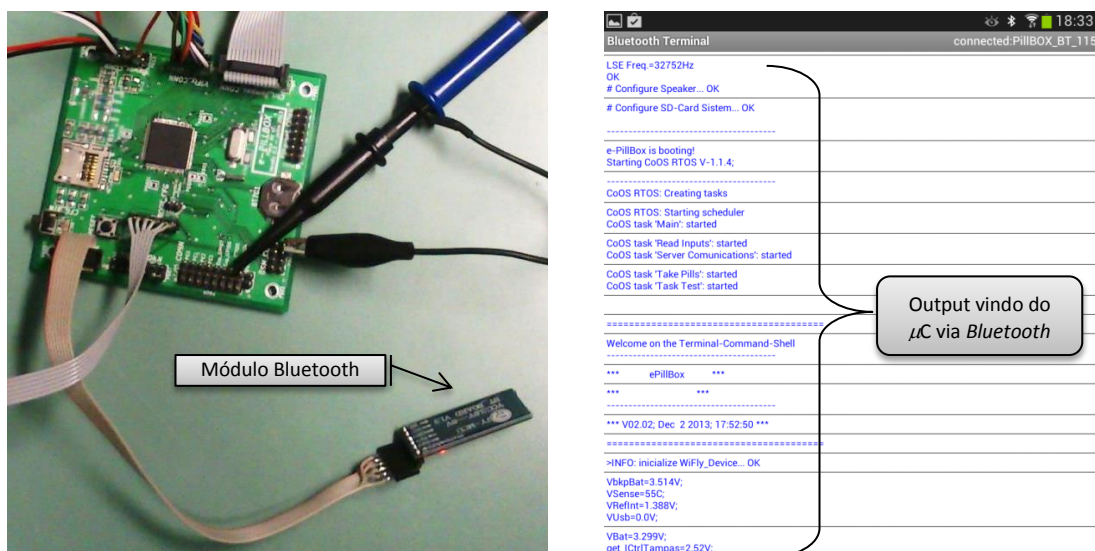


Figura 6.10 – Fotografia do módulo Bluetooth ligado ao μC e a imagem do programa XXX a mostrar os comandos enviados ao μC e as respostas enviadas de volta para o PC/tablet.

²³ O programa “Bluetooth Terminal” é um *software* de emulação de terminal para o Android e pode ser obtido através do site: <https://play.google.com/store/apps/details?id=Qwerty.BluetoothTerminal>

6.2.3.2 - Teste do módulo Ethernet

Este módulo, uma vez que foi adquirido, não necessita de grandes testes a nível de *hardware*. Os testes realizados têm como objetivo verificar o correto funcionamento de toda a pilha TCP/IP implementada no projeto. O teste consiste em colocar um servidor *web* rudimentar a funcionar e comutar um LED na placa de desenvolvimento. O objetivo foi bem conseguido com se pode ver pela figura 6.11.

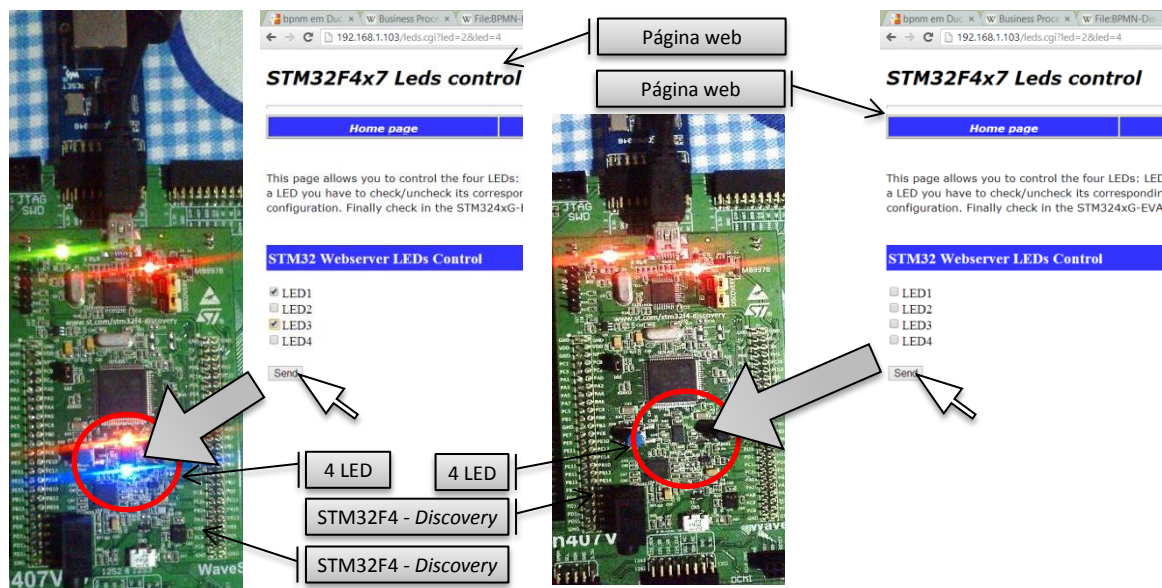


Figura 6.11 – Imagem da página Web e o estado do LED com conformidade com o que foi especificado na página.

6.2.3.3 - Teste do módulo Wi-Fi

Este módulo possui dois modos de funcionamento distintos:

1. Modo de configuração da *interface* de rede (Introdução do SSID da rede sem fios e palavra passe);
2. Estabelecimento de uma ligação ao servidor remoto para troca de dados.

Para testar o primeiro modo foi desenvolvido um programa que fazia o módulo *Wi-Fi* entrar em modo de configuração ao pressionar o botão “CFG_BTN” do módulo do μC .

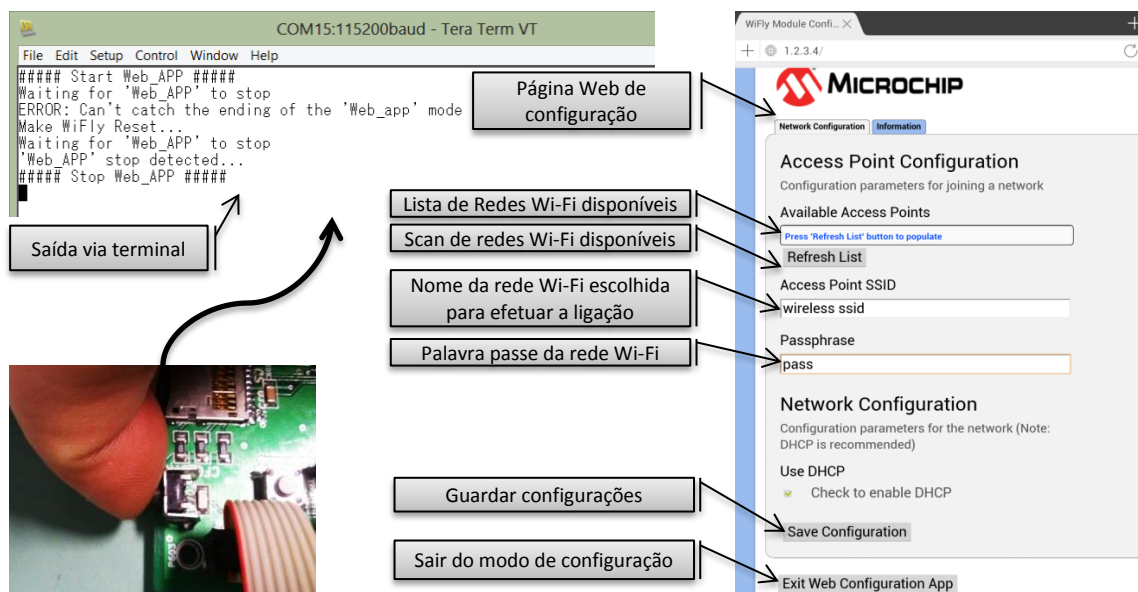


Figura 6.12 – Demonstração do teste de entrada em modo de configuração, configuração através de uma página web e terminar as configurações.

Para testar o segundo modo foi desenvolvido um programa que faz com que, ao pressionar o botão “CFG_BTN”, do módulo do μC , o módulo *Wi-Fi* estabeleça uma ligação ao servidor *Web* e mostre a resposta no terminal.



Figura 6.13 – Demonstração da comunicação com o servidor web através do módulo *Wi-Fi*.

Como se pode ver pelas figuras, o módulo funciona corretamente com o *software* desenvolvido.

6.2.4 - Leitura e escrita no cartão *MicroSD* e conversão de sinais analógicos

Neste caso, para testar esta parte foi desenvolvido um pequeno programa que, de segundo a segundo, recolhe uma amostra de todos os valores analógicos com interesse no projeto e armazena-os, juntamente com a data e hora, num ficheiro de texto no cartão *microSD* com sistema de ficheiros *FAT*. Os valores analógicos a ler são:

- Valores externos ao μC :
 - Tensão V_{USB} ;
 - Tensão V_{BAT} ;
 - Corrente I_{Ctrl_Tamas} .
- Valores internos ao μC :
 - Sensor de temperatura;
 - Tensão de referência interna;
 - Tensão I_{Bkp_BAT} ;

Os dados obtidos estão na Tabela 6.1.

Tabela 6.1 – Valores obtidos do ficheiro de leituras das amostras das entradas analógicas.

Data/hora	Tensão V_{USB}	Tensão V_{BAT}	Corrente I_{Ctrl_Tamas}	Sensor de temperatura;	Tensão de referência interna	Tensão I_{Bkp_BAT}
2013/11/31 23:00:05	4,9V	3,3	0A	12°	1,386V	3,507V
2013/11/31 23:00:06	5V	3,2	0A	12°	1,389V	3,50V
2013/11/31 23:00:07	4,7V	3,2	0A	12°	1,384V	3,45V
2013/11/31 23:00:08	4,9V	3,3	0A	12°	1,386V	3,55V

Como se pode ver pela semelhança entre os valores lidos pelo ADC e os registados no ficheiro, depreende-se que também esta parte do *software* está a funcionar com sucesso.

Capítulo 7 - O design da ePillbox

Este capítulo aborda a *conceção* do corpo do equipamento. A conceção do equipamento consistiu na criação da parte exterior e interior do equipamento. A “parte exterior” do equipamento é importante, já que é a sua parte mais visível. A “parte interior”, apesar de estar escondida, é a que permite que o equipamento consiga funcionar corretamente. O conteúdo deste capítulo foi elaborado com base nos textos e desenhos feitos pelo *designer* do grupo de trabalho – Rui Dias, aluno finalista da Escola Superior de Educação de Coimbra (ESEC), [88, pp. 55-81].

7.1 - Enquadramento

O *design* do dispositivo *ePillbox* visa uma aparência agradável e minimalista, na simplicidade de manipulação, tendo em consideração as especificidades dos seus utilizadores, em última análise, a tentar minimizar a necessidade de algum grau de literacia em saúde, tanto da parte do paciente ou do prestador de cuidados.

O *ePillbox* será extremamente fácil de usar, quer seja na toma de medicação ou na programação dos horários da medicação, que será feita previamente com o uso de uma página web.

O dispositivo utiliza vários meios de comunicação embutidos no equipamento, e foram projetados para um uso simples e eficiente, pelos diversos utilizadores que constituem seu público-alvo. Na figura 7.1 estão presentes os conceitos das formas da *ePillBox* que foram apresentados e discutidos pela equipa. O conceito escolhido pela equipa foi o número 3.

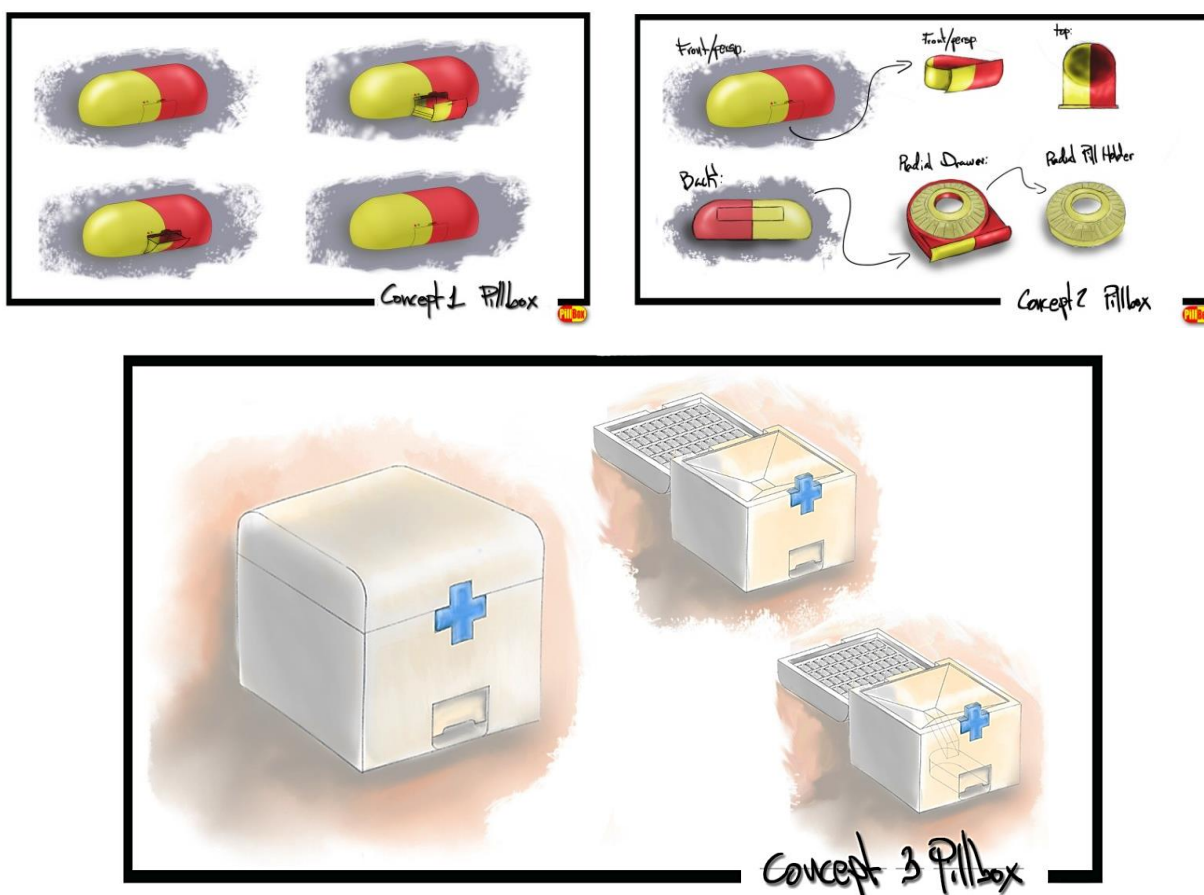


Figura 7.1 – Imagem do conceito do dispositivo proposto.

7.2 - O exterior da ePillbox

Tendo em conta a análise dos esboços anteriores relativamente às suas vantagens e desvantagens, e que independentemente do tipo de funcionamento que acabaríamos por optar deveria manter-se o conceito de apenas permitir o acesso dos medicamentos (por parte dos doentes) recorrendo à utilização de uma gaveta.

Este conceito conduziu a que se prosseguisse com a utilização da função inicial de aplicar um sensor à gaveta, de forma a registar se o doente tem seguido o seu tratamento com normalidade, notificando o prestador de cuidados, caso isso se não verifique.

Do ponto de vista do doente, o dispositivo aparenta um formato de baú (ver figura 7.2), que estaria sempre fechado (o acesso feito através de uma chave da qual o prestador de cuidados estaria responsável) e funcionaria da seguinte forma: um sinal luminoso e sonoro chamará a atenção para a toma da medicação; seguidamente, o doente terá de carregar num botão facilmente identificável no dispositivo; após isso, retira a gaveta de forma a ter acesso à medicação e proceder à toma dos mesmos; e finalmente, coloca a gaveta na sua posição de origem.

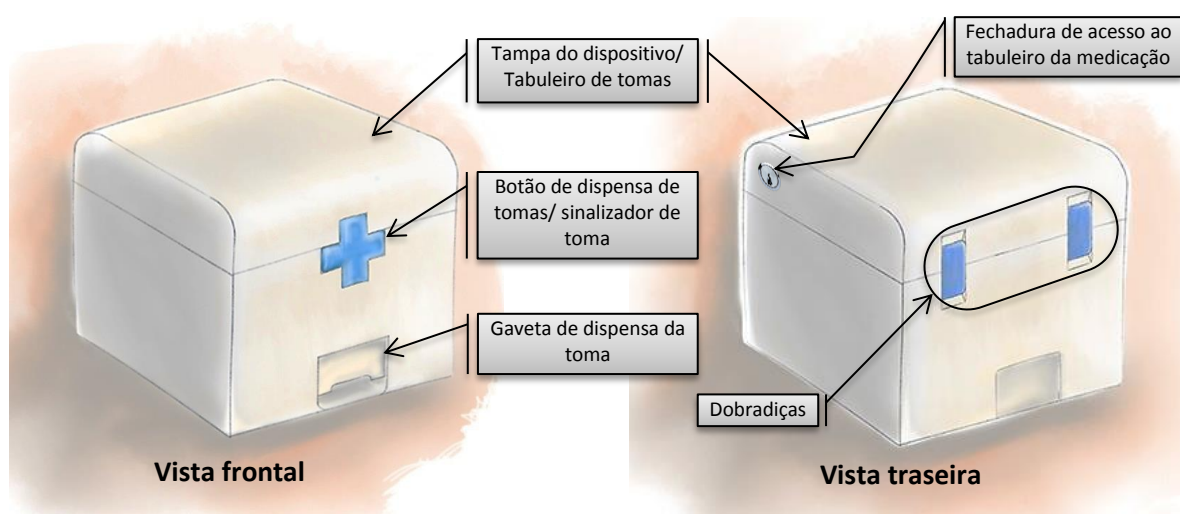


Figura 7.2 – Vista em prestativa da caixa fechada (vista frontal e traseira).

Com estes simples passos pode reduzir-se imenso a possibilidade de eventuais erros de utilização que provocariam interrupções no tratamento que teriam de ser retificados pelo prestador de cuidados antes de resumir as suas funções normais, tornando o dispositivo o mais autónomo e o mais seguro possível.

Referindo as suas funcionalidades, de um ponto de vista técnico, pode mencionar-se que com a colocação dos compartimentos no interior e na tampa do baú podem resolver-se alguns problemas, como, por exemplo, impossibilitar o acesso indevido aos medicamentos e fazer com que o mecanismo de abertura das tampas não necessite de proteção individual — visto que esta fica com uma fechadura que controlará o acesso ao interior da caixa, tornando-o bastante mais simples e menos dispendioso.

7.3 - O interior da ePillBox

Em dispositivos deste género sendo este o sistema matricial em que os compartimentos estão dispostos em formato de grelha, cada um com uma tampa individual de forma a impossibilitar a mistura de medicamentos nas diversas tomas e facilitar o processo de carregamento efetuado pelo prestador de cuidados sendo que cada compartimento se encontra identificado pelo dia da semana e período do dia, respetivamente.

Conforme o exposto no capítulo 2 - Estado da Arte, existem versões com e sem proteção, para eventuais tentativas de forçar a abertura de um compartimento errado. As versões com proteção, geralmente são do

tipo de equipamento menos acessível em termos de custo, possivelmente pelo tipo de mecanismos utilizados nas tampas, que não só podem ser acionados eletronicamente como também possuem força suficiente, ao ponto de dificultarem possíveis tentativas de se forçar a abertura de compartimentos errados ou fora do seu período de toma.

Na figura 7.3 está um plano da caixa aberta, onde se pode ver a localização do tabuleiro das tomas e dos compartimentos de tomas individuais. Neste caso, ao abrir a caixa, o tabuleiro das tomas fica visível e disponível para fazer o respetivo carregamento. Cada compartimento está equipado com uma tampa com a abertura controlada eletronicamente. Depois do tabuleiro corretamente carregado é necessário fechar as respetivas tampas manualmente, já que só a sua abertura é que é controlada eletronicamente. O próximo passo consiste então em fechar o tabuleiro das tomas. Depois de fechado, as tampas funcionam como alçapões. A tampa e o respetivo mecanismo de ativação têm que ser suficientemente fortes para suportar o peso dos medicamentos.

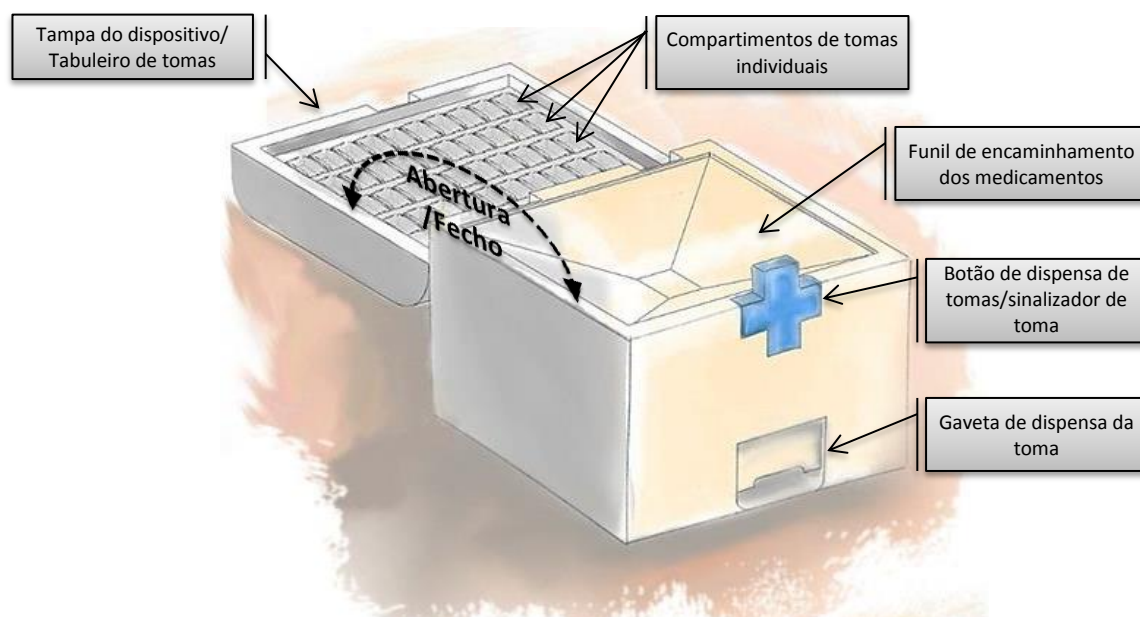


Figura 7.3 – Vista da caixa aberta com detalhes do compartimento do tabuleiro.

Desta forma, o único cuidado a ter dá-se enquanto a tampa do baú se encontra fechada (ficando os compartimentos numa posição invertida), e o mecanismo de fecho das tampas suportar o peso da própria tampa, juntamente com os medicamentos que poderão estar contidos nesse compartimento.

Todavia, chegado o momento/horário da toma do medicamento, o doente ao carregar no botão irá confirmar a sua intenção de tomar a medicação, fazendo com que o mecanismo seja acionado eletronicamente, de forma a destrancar a tampa do compartimento, permitindo a abertura desta através da força de gravidade e, ao mesmo tempo, os medicamentos, que também estão sujeitos à mesma gravidade, caírem numa espécie de funil embutido na secção inferior do baú, sendo, então, como que empurrados para a gaveta respetiva, para que seja possível proceder-se à toma dos mesmos. Na figura 7.4 pode ver-se o pormenor do interior do equipamento.

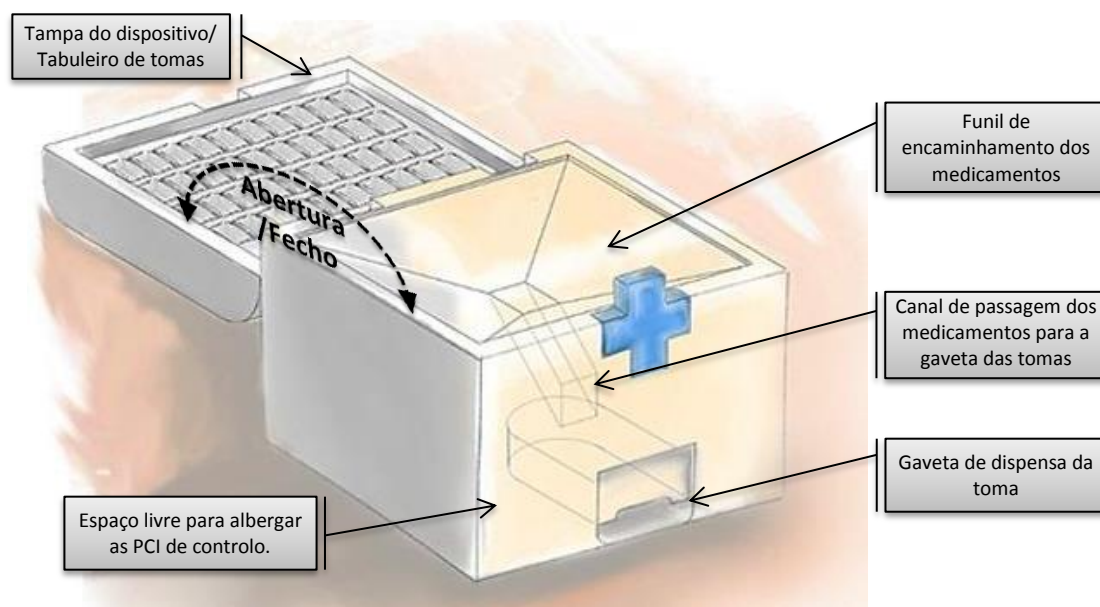


Figura 7.4 – Vista com detalhes sobre o interior da secção inferior do dispositivo.

Ao enveredar por este tipo de *design* é dada também a possibilidade de se utilizar os espaços vazios da secção inferior — entre as partes laterais do baú e a parte onde se encontra o funil e a gaveta —, para a colocação do *hardware* necessário, contendo espaço suficiente para que possam ser utilizados vários tipos de placas de processamento, não se ficando limitado a um espaço reduzido.

7.3.1 - Os compartimentos individuais

O tabuleiro das tomas é constituído por 35 compartimentos individuais com tampa e mecanismo de abertura. Na figura 7.5 pode ver-se um pormenor do compartimento individual de tomas e do respetivo mecanismo de abertura da tampa. O que está presente na figura poderá não ser a versão final, já essa parte ainda está em estudo e, como tal, por ser sujeita a alterações.

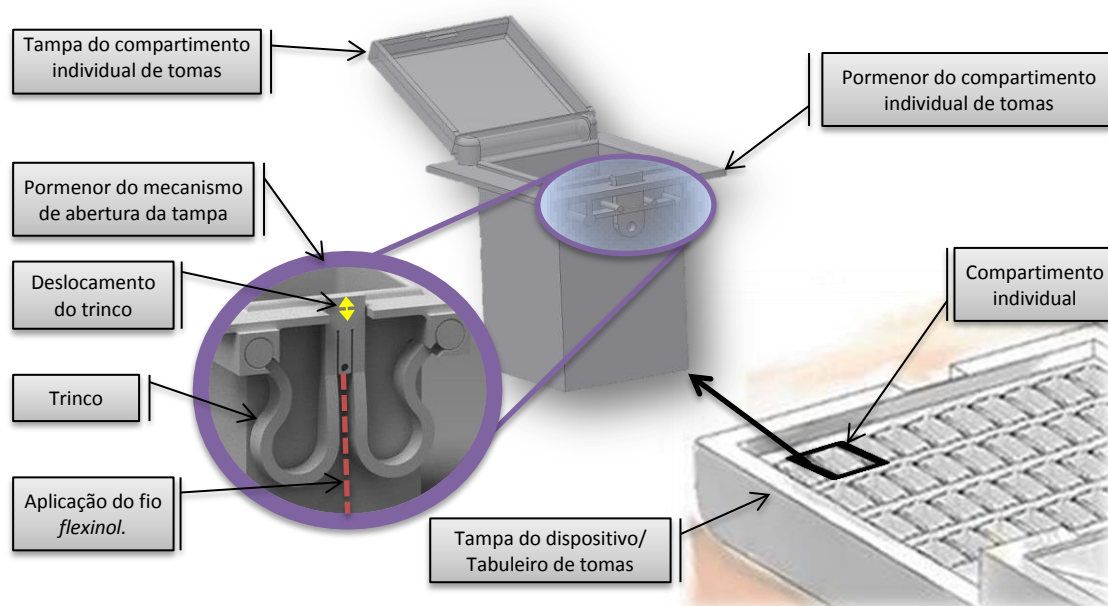


Figura 7.5 – Pormenor do compartimento individual de tomas e do mecanismo de abertura da tampa.

7.3.2 - Mecanismo de abertura das tampas dos compartimentos

O mecanismo de abertura das tampas dos compartimentos individuais, destinados ao armazenamento de medicação, é um dos aspetos mais importantes e de maior dificuldade no desenvolvimento do dispensador.

Definiu-se uma abordagem de construção de dentro para fora, em que os problemas fossem abordados e resolvidos de forma linear consoante forem surgindo.

Inicialmente foi sugerida a utilização de um dispositivo transdutor solenoide. No entanto, a sua implementação neste dispositivo iria revelar-se como uma opção pouco viável em termos financeiros, já que se iria necessitar de um atuador por compartimento, o que perfaz um total de 35 unidades. Outro dos inconvenientes foi o seu tamanho físico que seria demasiado grande para caber no espaço entre os compartimentos sem se ter que aumentar o tamanho do tabuleiro. Outra alternativa surgida foi de atuadores piezoelétricos, o que resolveria o problema do tamanho; no entanto apresentava duas limitações: a primeira tem a ver com o pouco curso que este tipo de atuador tem, [89]; a segunda tem ver com o custo unitário que é superior ao dos solenoides.

O tipo de atuador escolhido foi o *flexinol* (*muscle wire*), tendo as suas características e funcionamento sido abordadas no ponto 4.4.1- na página 39.

O funcionamento do trinco é muito simples e consiste no seguinte (o ponto de partida é a tampa fechada):

- Inicia-se a passagem de uma corrente elétrica pelo *flexinol*, o que faz com que este contraia;
- Ao contrair puxa o trinco no sentido do fundo do copo, libertando assim a tampa;
- Como os compartimentos estão invertidos, os medicamentos estão em cima da tampa e, pela ação da gravidade, faz com que esta abra, libertando assim os medicamentos;
- Após um determinado tempo, a corrente para de passar e faz com que o *flexinol* arrefeça. Ao arrefecer dilata, e com a ajuda da mola natural do trinco volta à posição inicial.

7.4 - Sumário

Neste capítulo foi abordada a conceção do equipamento. Foram indicados os diversos esboços até à escolha da configuração final. Também foram ilustrados os aspetos exterior e interior, bem como a localização dos diversos elementos que o compõem. Por fim foi explicitada a forma de controlo das tampas dos compartimentos.

Capítulo 8 - Conclusões

Este capítulo apresenta as conclusões que se podem retirar do projeto realizado e apresenta sugestões para trabalhos futuros a partir da plataforma desenvolvida.

8.1 - Conclusões

A presente monografia teve como objetivo descrever o desenvolvimento de um dispensador automático de medicamentos. Nessa medida, o presente projeto reúne um levantamento dos equipamentos já presentes no mercado e uma comparação entre as suas características. No desenvolvimento tentou-se incorporar características que diferenciasssem este equipamento dos outros já existentes no mercado. São exemplo disso a capacidade de incorporação de diversos meios de comunicação, tabuleiro com 35 tomas (é equivalente a 5 tomas diárias durante 7 dias), e um *design* inovador.

A solução desenvolvida vai ao encontro dos objetivos propostos inicialmente. Desenvolveu-se um protótipo de uma solução microcontrolada, com capacidade para gerir a entrega atempada da medicação aos pacientes. Foi criada uma plataforma física e virtual que suporta comunicação por diversos meios de comunicação (*GSM*, *Wi-Fi*, *Bluetooth*, *Ethernet*), com RTCC, com capacidade para armazenar e ler informação localmente, e bateria de alimentação para manter o dispositivo funcional em caso de falha da fonte de alimentação principal. A nível virtual foi criada uma página *web* para: administrar o equipamento remotamente; enviar alertas aos prestadores de cuidados no caso de falha da toma; e visualizar os relatórios de cumprimento do plano de tomas.

Neste momento, o equipamento pode funcionar como plataforma de testes e desenvolvimento de funcionalidades extras, já que possui uma arquitetura modular. Também o μC utilizado é relativamente recente e possui potencialidades extras, a nível de capacidade de processamento e periféricos internos, que permitem desenvolver novas funcionalidades, tanto a nível de *software* como de *hardware*. Tendo em conta o facto do equipamento vir a ser comercializado, foram implementados meios na PCI para permitir a troca do μC atual por um mais simples e barato.

Outro dos pontos relevantes é que todo o *hardware* (desde a captura de esquemático e desenho de PCI), foi realizado com ferramentas *Open Source*. As ferramentas utilizadas no desenvolvimento do *software* também são ferramentas disponíveis sem custos e com licenças compatíveis com *Open Source*. O próprio *software* desenvolvido também incorpora bibliotecas de várias pessoas/organismos com licenças abertas e que permitem a sua inclusão em produtos comerciais. Com isto se prova que, atualmente já é possível realizar projetos deste nível sem recurso a ferramentas proprietárias.

Por último, mas não menos importante, é de salientar que este projeto faz parte de uma tendência que já é muito seguida no mundo empresarial, mas que no meio académico ainda é fraca, e que corresponde ao facto de envolver várias áreas do conhecimento (média, eletrónica, *software*, mecânica e *design*) e instituições. Este tipo de projetos em que se criam sinergias de pessoas/instituições com domínios de conhecimento e recursos diversos faz com que se criem projetos mais completos e que respondam às necessidades reais das pessoas. No entanto, também traz novos desafios no que toca à troca de informação, já que não se pode utilizar uma linguagem tão técnica pois pode correr-se o risco de não se conseguir passar a mensagem.

Finalmente, e como diz o povo, “a prática faz a perfeição”, neste caso não é diferente; pelo que, com a continuação de projetos deste género, é possível criar estratégias e metodologias que façam com que o sucesso deste tipo de projetos aumente significativamente.

8.2 - Trabalhos futuros

Dado que o equipamento ficou operacional, mas não está completamente terminado, quer em termos de *software*, *hardware* e estrutura, é necessário terminar os pontos que faltam para se ter um protótipo completamente funcional e que possa ser utilizado por um paciente.

A próxima fase consistiria no desenvolvimento de algumas unidades, o mais próximo possível do produto final, para realizar um ensaio com vários pacientes de diferentes idades e limitações. Isto permitiria “limar as arestas do projeto”, desde a página *web* até ao próprio funcionamento do equipamento.

De seguida serão discriminados os pontos a melhorar na aplicação atual e/ou funcionalidades a implementar.

8.2.1 - Melhorias no *firmware*;

8.2.1.1 - Sistema de Comunicações

- Aplicação de um método de compressão de dados para reduzir os encargos em meios de comunicação, sendo que o custo varia com a quantidade de dados transmitidos/recebidos;
- Implementação de um método de encriptação das mensagens trocadas entre o equipamento e o servidor *web*. O método deve garantir duas vertentes:
 - i. A segurança do conteúdo das mensagens para preservar a privacidade dos pacientes;
 - ii. A autenticidade do conteúdo da mensagem e dos intervenientes da comunicação;
- No caso da comunicação *Wi-Fi* incorporar um servidor *web* para configuração do equipamento, substituindo assim a página do módulo *Wi-Fi* que não é personalizável e possui demasiados parâmetros disponíveis que podem, inclusivamente, deixar o módulo *Wi-Fi* inoperacional;
- Desenvolver uma aplicação para *smartphones*, *tablet* ou *PC*, para interagir com o dispositivo através de *Bluetooth* ou um dos outros meios disponíveis no *ePillBox*. A aplicação poderá servir de *interface* direta com o equipamento ou interagir com o servidor *web* utilizando *webservices*.
- Desenvolvimento de drivers que permitam a utilização de um modem GSM em vez do modem *Wi-Fi* que atualmente integra o projeto.

8.2.1.2 - Geral

- Implementação de um módulo de áudio capaz de reproduzir ficheiros de áudio armazenados num cartão *MicroSD*. Integrando-lhe um microfone, o dispositivo poder-se-á transformar num meio de comunicação entre o paciente e o prestador de cuidados. Assim, o paciente, quando desejasse, poderia falar diretamente com o prestador de cuidados através do equipamento.
- Implementar algoritmos de deteção de desgaste da bateria de *backup* para estimar a sua vida útil e sugerir a sua substituição.

8.2.2 - Melhorias no *hardware*

A nível do *hardware* é possível realizar algumas melhorias no seu desenho e escolha de componentes, para construir um equipamento com uma única PCI, mais pequena que a atual e mais amigável para produção automatizada.

Dado que o módulo da fonte de alimentação não ficou operacional é necessário averiguar a causa e refaze-lo de novo. A solução pode passar por conceber uma nova fonte de alimentação recorrendo ao módulo em [87] em vez de tentar desenvolver um conversor DC-DC de raiz.

Outra das partes a melhorar será a parte do carregador de baterias. Será necessário desenvolver um que permita efetuar a carga e manutenção de baterias do tipo NI-MH ou de Lítio ou mesmo de ambas, aumentando assim a versatilidade deste mesmo módulo.

A nível de funcionalidades poder-se-ia incorporar sensores, como, por exemplo:

- **Acelerómetro** – Poderia servir para detetar deslocamentos do dispositivo ou quedas, por exemplo;
- **Sensor de temperatura e humidade** – Permitiria verificar se os medicamentos não sofreram alterações devido à exposição a fatores ambientais adversos.

8.2.3 - Melhorias no servidor remoto (páginas web e base de dados)

Relativamente ao servidor de página *web*, as sugestões de trabalho futuro são:

- Implementação do protocolo HTTPS obrigatório no acesso à página *web* aumentando assim o nível de confidencialidade dos conteúdos trocados entre o servidor remoto e os navegadores dos utilizadores;
- Disponibilização de *web services* para facilitar a troca de informação entre o servidor *web* e outras aplicações.

Referências bibliográficas

- [1] Siddall, Clare; Kjaeserud, Guri; et al., "Healthy ageing, a keystone for a Sustainable Europe-EU Health Policy in the Context of Demographic Change," *Health & Consumer Protection*, Janeiro 2007.
- [2] M. J. Carrilho e L. Patrício, "A Situação Demográfica Recente em Portugal," *Revista de Estudos Demográficos* - Nº36, pp. 127-152, 2004.
- [3] M. J. Carrilho e M. d. L. Craveiro, "A Situação Demográfica Recente em Portugal," *Revista de Estudos Demográficos* - Nº50, pp. 45-91, 2013.
- [4] S. Lagarto, C. Nunes, D. Gomes e M. F. Mendes, "A evolução na mortalidade da população idosa portuguesa, por sexos e por causas de morte: Identificação de aglomerados espaço-temporais por NUTSIII," *Revistas de Estudos Demográficos*, Nº50, pp. 5-30, 2013.
- [5] Fundação Francisco Manuel dos Santos – PORDATA, "Indicadores de envelhecimento segundo os Censos em Portugal," 2013. [Online]. Available: <http://www.pordata.pt/Portugal/Indicadores+de+envelhecimento+segundo+os+Censos-525>. [Acedido em 25 Março 2013].
- [6] Kutsal, Yesim Gokce et al., "Polypharmacy in the Elderly: A Multicenter Study," *American Medical Directors Association*, 2009.
- [7] Sousa, Salete et al., "Polimedicação em doentes idosos: adesão à terapêutica," *Revista Portuguesa de Clínica Geral*, 2011.
- [8] J. A. Erlen e J. A. Erlen, "Functional health illiteracy. Ethical concerns," *Orthop Nurs*, 23(2), pp. 150-153.
- [9] Y. V. Martínez e C. A. e. a. Prado-Aguilar, "Quality of life associated with treatment adherence in patients with type 2 diabetes: a cross-sectional study," *BMC Health Services Research*, p. 8:164, 2008.
- [10] T. G. Heloisa, L. Z. Maria e J. H. Vanderlei, "Factors Related to Patient Adherence to Antidiabetic Drug Therapy," *Ver. Latino – Am Enfer. January-February*, pp. 46-51, 2009.
- [11] N. Vinagre, R. Cruz, I. Fonseca e J. T. Farinha, "Dispensador individual eletrónico de medicamentos - ePillbox," *Tecno hospital* Nº 57, pp. 26-31, Maio/junho 2013.
- [12] M. Hernández e J. L. Poveda, *Sistemas Automaticos de Dispensacion de Medicamentos*, 78. 08028 Barcelona: Dispensación de medicamentos. Coordinación: Joaquim Bonal. Dirección: M.ª Cinta Gamundi Planas. Impreso por Press Line - Juan Güell, 2001.
- [13] Med-E-Lert.net, "Medelert Automatic Pill Dispenser," 2013. [Online]. Available: http://medelert.net/index.php?route=product/product&path=59&product_id=50. [Acedido em 9 Agosto 2013].

- [14] TabTime.com, "Tabtime Medelert: Automatic-lockable-timed release pill dispenser," 2008. [Online]. Available: <http://www.tabtime.com/medelert.html>. [Acedido em 9 Agosto 2013].
- [15] MedReady Inc., "Medication Dispensers," 2013. [Online]. Available: http://metroid.heritagewebdesign.com/~anthony2/cart/index.php?main_page=index&cPath=6. [Acedido em 9 Agosto 2013].
- [16] E-pill.com, "Automatic Pill Dispensers," 2011. [Online]. Available: <http://www.epill.com/dispenser.html>. [Acedido em 2013 Abril 2013].
- [17] Addoz, "Addoz oy website," 2013. [Online]. Available: <http://www.addoz.com/home>. [Acedido em 5 Agosto 2013].
- [18] PHILIPS, "Philips Medicamentation Dispensing Service," [Online]. Available: <http://www.managemypills.com/content/product-details>. [Acedido em 29 Abril 2013].
- [19] MedMinder, "Automatic Pill Dispenser," 2012. [Online]. Available: <http://www.medminder.com>. [Acedido em 2013 Abril 2013].
- [20] MedicPen, "MedicPen website," 2013. [Online]. Available: <http://www.medicpen.se>. [Acedido em 5 Agosto 2013].
- [21] E-pill.com, "Portable Automatic Pill Dispenser," 1 Agosto 2013. [Online]. Available: <http://www.epill.com/medimi.html>. [Acedido em 6 Agosto 2016].
- [22] STMicroelectronics, "STM32F407VG Datasheet - Rev. 4," 1 julho 2013. [Online]. Available: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00037051.pdf>.
- [23] J. Yiu, The definitive guide to the ARM® Cortex-M3, Second Edition, Elsevier, 2010.
- [24] Mouser Electronics, Inc., "STM32F4DISCOVERY STMicroelectronics | Mouser," 2013. [Online]. Available: <http://pt.mouser.com/ProductDetail/STMicroelectronics/STM32F4DISCOVERY/?qs=sGAEpiMZZMutVogd4PRsvEN8XDBeCtgD>. [Acedido em 5 Dezembro 2013].
- [25] "Web site do KiCad EDA Software Suite," Abril 2013. [Online]. Available: <http://www.kicad-pcb.org/display/KICAD/KiCad+EDA+Software+Suite>.
- [26] M. Brown, Power Supply Cookbook - Second Editon, United States of America: Newnes, 2001.
- [27] Linear Technology, "LTC3536 Datasheet," 15 Abril 2013. [Online]. Available: <http://www.linear.com/docs/40104>.
- [28] D. Linden, "CHAPTER 1 - Basic Concepts," em *Handbook of batteries - Third Edition*, McGraw-Hill, 2002, pp. 1.3-1.18.
- [29] Wikipedia, "Battery (electricity)," Wikipedia, 2013. [Online]. Available: [https://en.wikipedia.org/wiki/Battery_\(electricity\)](https://en.wikipedia.org/wiki/Battery_(electricity)). [Acedido em 3 Dezembro 2013].
- [30] B. Chu, AN1088 - Selecting the Right Battery System For Cost-Sensitive Portable Applications While Maintaining Excellent Quality, Microchip Technology Inc., 2007.

-
- [31] Linear Technology Corporation, "Datasheet:LTC4075/LTC4075X - Dual Input USB/AC Adapter Standalone Li-Ion Battery Chargers," 2005. [Online]. Available: <http://www.linear.com/docs/7878>.
- [32] Microchip Technology Incorporated, "Datasheet: MCP73833/4 - Stand-Alone Linear Li-Ion / Li-Polymer Charge Management Controller," 2009. [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/22005b.pdf>.
- [33] Texas Instruments Incorporated, "Datasheet: bq2000 - Programmable Multi-Chemistry Fast-Charge Management IC," 2010. [Online]. Available: <http://www.ti.com/lit/ds/symlink/bq2000.pdf>.
- [34] Linear Technology, "LTC4054-4.2 - Standalone Linear Li-Ion Battery Charger with Thermal Regulation in ThinSOT," 2003. [Online]. Available: <http://www.linear.com/docs/2288>.
- [35] STMicroelectronics, "STM32F4DISCOVERY - User Manual," 10 fevereiro 2013. [Online]. Available: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00039084.pdf.
- [36] USB Implementers Forum, "11 - Interoperability and Power Delivery," em *Universal Serial Bus 3.0 Specification, Revision 1.0 (including errata and ECNs through May 1, 2011)*, 2011, pp. 1-11.
- [37] STMicroelectronics, "Reference manual - STM32F405xx/07xx, STM32F415xx/17xx, STM32F42xxx and STM32F43xxx advanced ARM-based 32-bit MCUs (Rev.5)," Setembro 2013. [Online]. Available: http://www.st.com/web/en/resource/technical/document/reference_manual/DM00031020.pdf. [Acedido em Setembro 2013].
- [38] STMicroelectronics, "AN2606 - STM32™ microcontroller system memory boot mode - rev.17," Maio 2013. [Online]. Available: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/application_note/CD00167594.pdf. [Acedido em Setembro 2013].
- [39] STMicroelectronics, "UM0412 - User manual - Getting started with DfuSe USB device firmware upgrade STMicroelectronics extension - Rev 4," julho 2009. [Online]. Available: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/CD00155676.pdf. [Acedido em Abril 2013].
- [40] Microchip Technology Inc., "MCP23008/M23S08 - 8-Bits I/O Expander with Serial Interface," Agosto 2007. [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/21919e.pdf>. [Acedido em Março 2013].
- [41] Comissão Europeia, "TECHNICAL ANNEX TO: MOU REGARDING THE HARMONISATION OF A CHARGING CAPABILITY FOR MOBILE PHONES," 12 Janeiro 2010. [Online]. Available: http://ec.europa.eu/enterprise/sectors/rte/files/chargers/chargers_annex_ii_to_mou_january_12_2010_en.pdf. [Acedido em Abril 2013].
- [42] R. Dunlop e A. Garcia, "A Nitinol Wire Actuated Stewart Platform," em *Australasian Conference on Robotics and Automation*, Auckland, Novembro de 2002.
- [43] N. C. Braga, "SMA - Músculos eletrônicos," *Mecatrônica Actual Nº1*, pp. 48-52, Outubro-Novembro 2001.
- [44] Images SI, Inc, "Nitinol - Shape Memory Alloy," 2007. [Online]. Available: <http://www.imagesco.com/articles/nitinol/01.html>. [Acedido em 20 Abril 2013].
-

- [45] DYNALLOY, Inc., “FLEXINOL® Actuator Wire Technical and Design Data,” 2013. [Online]. Available: <http://www.dynalloy.com/pdfs/TCF1140.pdf>. [Acedido em 2 Abril 2013].
- [46] Roving Networks, Inc., “RN-171-XV Data Sheet,” 10 Outubro 2012. [Online]. Available: <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en560635>. [Acedido em 19 Abril 2013].
- [47] Multi-Tech systems, “Global SocketModem GSM/GPRS Developer’s Guide - MTSMC-G-F1 – GSM/GPRS Class 10, 900/1800 MHz (Rev. A),” 16 Maio 2003. [Online]. Available: <http://www.Multi-Tech.com>. [Acedido em 15 Abril 2013].
- [48] Philips Semiconductors, “AN97055 - Bi-directional level shifter for I²C-bus and other systems.,” 1997. [Online]. Available: http://www.nxp.com/documents/application_note/an97055.pdf. [Acedido em 30 Setembro 2013].
- [49] Guangzhou HC Information Technology Co., Ltd., “DataSheet do HC-06 - rev.2.2,” 6 Abril 2011. [Online]. Available: http://iw.suntekstore.com/office_cache/265/14004375/132885991714004375.pdf. [Acedido em 10 janeiro 2013].
- [50] Waveshare International Ltd., “DP83848 Ethernet Board (Esquemático),” 2013. [Online]. Available: <http://www.wvshare.com/product/DP83848-Ethernet-Board.htm>. [Acedido em 1 Abril 2013].
- [51] Institute for Interconnecting and Packaging Electronic Circuits, “2221B: Generic Standard on Printed Board Design,” Novembro 2012. [Online]. Available: https://portal.ipc.org/Purchase/ProductDetail.aspx?Product_code=db07d393-2133-e211-bf0f-00155d052876. [Acedido em 1 Junho 2013].
- [52] D. L. Jones, “PCB Design Tutorial - Rev. A,” 29 junho 2004. [Online]. Available: <http://www.alternatezone.com/electronics/files/PCBDesignTutorialRevA.pdf>. [Acedido em 17 Abril 2013].
- [53] Altron Inc., “Board Design Guidelines,” 2003. [Online]. Available: <http://www.altronmfg.com/files/Board%20Design%20Guidelines%202003%20Rev-A.pdf>. [Acedido em 25 Abril 2013].
- [54] Wikipédia, “Firmware,” 1 Novembro 2013. [Online]. Available: <https://en.wikipedia.org/wiki/Firmware>. [Acedido em 1 Novembro 2013].
- [55] Red Hat, inc, “The newlib homepage,” [Online]. Available: <https://sourceware.org/newlib/>. [Acedido em 23 Fevereiro 2013].
- [56] Free Software Foundation, Inc., “lwIP - A Lightweight TCP/IP stack - Resumo,” 2013. [Online]. Available: <http://savannah.nongnu.org/projects/lwip/>. [Acedido em 22 Abril 2013].
- [57] W. Bober, “wbober / jsnm — Bitbucket,” 2013. [Online]. Available: <https://bitbucket.org/wbober/jsnm>. [Acedido em 16 Agosto 2013].
- [58] ChaN, “FatFs - Generic FAT File System Module,” 2 Outubro 2013. [Online]. Available: http://elm-chan.org/fsw/ff/00index_e.html. [Acedido em 1 Novembro 2013].
- [59] UB, “FATFS_SDIO library (STM32F4),” [Online]. Available: http://mikrocontroller.bplaced.net/wordpress/?page_id=621. [Acedido em 15 Agosto 2013].

-
- [60] T. Noergaard e J. Labrosse, “Capítulo 3 - Embedded Operating Systems,” em *Embedded Software - Know It All*, Burlington - MA, Elsevier, 2008, pp. 169 - 239.
- [61] Wikimedia Foundation, Inc., “List of real-time operating systems,” [Online]. Available: https://en.wikipedia.org/wiki/List_of_real-time_operating_systems. [Acedido em 15 Abril 2013].
- [62] Coocox, “Free and open ARM Cortex MCU RTOS,” 2011. [Online]. Available: <http://www.coocox.org/CoOS.htm>. [Acedido em 15 Abril 2013].
- [63] Coocox, “Coocox CoOS User's Guide - Revision 1.1,” Agosto 2009. [Online]. Available: http://www.coocox.org/downloadfile/CoOS/PDF/Coocox_CoOS_User_Guide.pdf. [Acedido em 15 Abril 2013].
- [64] T. Waldock, “Arduino WiFly Driver,” 2013. [Online]. Available: <http://sourceforge.net/projects/arduinowifly/files/>. [Acedido em 23 Março 2013].
- [65] JSON-RPC Working Group, “JSON-RPC 2.0 Specification,” 4 Janeiro 2013. [Online]. Available: <http://www.jsonrpc.org/specification>. [Acedido em 15 Setembro 2013].
- [66] B. Meyer, “UML: The Positive Spin,” [Online]. Available: <http://archive.eiffel.com/doc/manuals/technology/bmarticles/uml/page.html>.
- [67] Uriel, “[dev] Suckless UML,” Maio 2011. [Online]. Available: <http://lists.suckless.org/dev/1105/7901.html>, e http://en.wikipedia.org/wiki/Unified_Modeling_Language.
- [68] “The Dot Language,” [Online]. Available: www.graphviz.org.
- [69] “GraphML – Graphical XML,” [Online]. Available: <http://graphml.graphdrawing.org/>.
- [70] “IFML and WebML and BPMN,” [Online]. Available: omg.org/spec/IFML, ifml.org, webml.org, www.bpmn.org e <http://www.omg.org/spec/BPMN/Current/>.
- [71] “Modelling languages and tools,” [Online]. Available: modeling-languages.com e papyrusuml.org, topcased.org, modelio.org, sourceforge.net/projects/whitestaruml e <http://www.sparxsystems.com.au/>.
- [72] “WebRatio – IFML tool,” [Online]. Available: <http://www.webratio.com>.
- [73] “Linguagens,” [Online]. Available: PHP-php.net, Haskell-haskell.org, Clojure-clojure.org, fsharp.org; groovy.codehaus.org.
- [74] “Frameworks – PHP,” [Online]. Available: [yii – yiiframework.com](http://yii-framework.com); [symfony – symfony.com](http://symfony.com); [kohana – kohanaframework.org](http://kohana-framework.org); [PPI – ppi.io](http://ppi.io), [phalcon – phalconphp.com](http://phalconphp.com), [FuelPHP – fuelphp.com](http://fuelphp.com), [ScriptCase – scriptcase.net](http://scriptcase.net);
- [75] “MVC methodology,” [Online]. Available: <http://www.javascriptmvc.com> e <http://backbonejs.org>.
- [76] “WebServices specification languages,” [Online]. Available: www.w3.org/TR/wSDL, www.w3.org/Submission/wadl/ ou <http://wadl.java.net/> e para o REST: <http://rest.elkstein.org/2008/02/documenting-rest-services-wsdl-and-wadl.html>.

- [77] "Web Client side technologies," [Online]. Available: en.wikipedia.org/wiki/Web_development, www.dartlang.org web.
- [78] "Client plus server side developing," [Online]. Available: pyjamas – pyjs.org, tarsus – www.tersus.com, CoffeeScript – coffeescript.org, Opa – opalang.org, TypeScript – www.typescriptlang.org;
- [79] "SQL," [Online]. Available: linq – msdn.microsoft.com/en-us/library/vstudio/bb397926.aspx, ORM – www.orm.net;
- [80] "Object-relational impedance mismatch," [Online]. Available: <http://marcorogers.com/blog/01-08-2009/why-you-should-be-using-an-object-database-instead-of-orm-part-1>, http://en.wikipedia.org/wiki/Object-Relational_impedance_mismatch.
- [81] "Frameworks – Python," [Online]. Available: Pylons – pylonsproject.org; TurboGears – turbogears.org; Zope – zope.org; django – djangoproject.com;
- [82] "Frameworks – Outras," [Online]. Available: Grails – grails.org, html5boilerplate – html5boilerplate.com, Twitter Bootstrap – twitter.github.com/bootstrap;
- [83] "JavaScript libraries," [Online]. Available: JQuery – jquery.com, MooTools – mootools.net, Prototype – prototypejs.org, Dojo – dojotoolkit.org, YUI – yuilibrary.com, ExtJS – sencha.com, AlloyUI – alloyui.com..
- [84] " PHP compilers," [Online]. Available: bcompiler- php.net/manual/en/book.bcompiler.php; hhvm- hhvm.com/blog; phc compiler- phpcompiler.org, Phalanger-phalanger-php-compiler.net.
- [85] Waveshare International Ltd., "STM32 QFP100 programming adapter test socket," 2013. [Online]. Available: <http://www.wvshare.com/product/STM32-QFP100.htm>. [Acedido em 1 Abril 2013].
- [86] Seeed Studio, "Seeed Studio Bazaar, Boost ideas, extend the reach - Fusion PCB," 2013. [Online]. Available: <http://www.seeedstudio.com/service/index.php?r=site/pcbService>. [Acedido em 12 Junho 2013].
- [87] Pololu Corporation, "Pololu 3.3V Step-Up/Step-Down Voltage Regulator S7V8F3," [Online]. Available: <http://www.pololu.com/product/2122>. [Acedido em 01 Dezembro 2013].
- [88] R. Dias, "Relatório de estágio para a obtenção do grau de Licenciado em: Comunicação e Design Multimédia (CDM)," Escola Superior de Educação de Coimbra - Departamento de Artes e Tecnologias, Coimbra, 2013.
- [89] M. Novotny e P. Ronkanen, "Piezoelectric Actuators," Tampere University of technology, Tampere , Finland.
- [90] Microchip Technology Inc., "WiFly Command Reference, Advanced Features and Applications User's Guide (RN-WIFLYCR-UG)," 30 Abril 2013. [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/50002230A.pdf>. [Acedido em 15 Maio 2013].

Anexos

Anexo A - Mapa da utilização dos pinos do microprocessador

Anexo B - Esquemas eletrónicos do projeto

B.1 - Interligação entre Módulos	B-1
B.2 - Esquema do módulo de atuação das tampas	B-2
B.3 - Esquema do módulo <i>Bluetooth</i>	B-3
B.4 - Esquema do módulo do μC	B-4
B.5 - Esquema do módulo GSM	B-6
B.6 - Esquema do módulo da fonte de alimentação	B-7
B.7 - Esquema do módulo <i>Ethernet</i>	B-8
B.8 - Esquema do módulo Wi-Fi	B-9

Anexo C - Desenho das Placas de Circuito Impresso

C.1 - Placa do Módulo do μC	C-1
C.2 - Placa do Módulo de controlo das tampas	C-3
C.3 - Placa do Módulo da fonte de alimentação	C-4
C.4 - Placa do Módulo de bluetooth	C-5

Anexo D - Listagem e descrição das funções da API do servidor web

D.1 - Listagem de procedimentos da API	D-1
--	-----

Anexo E - Parametrização do módulo Wi-Fi (RN-171-XV WiFly)

E.1 - Introdução	E-1
E.2 - Esquemático do módulo	E-1
E.3 - Listagem de parâmetros alterados	E-1
E.4 - Listagem de comandos para alterar os parâmetros	E-2
E.5 - Referências	E-3

Anexo A - Mapa da utilização dos pinos do microprocessador

Nº do pino	Nome do pino	Função do pino	Nome da ligação
1	PE2	Entrada/Saída digital Nº2	PE2
2	PE3	Entrada/Saída digital Nº3	PE3
3	PE4	Entrada/Saída digital Nº4	PE601
4	PE5	Entrada/Saída digital Nº5	PE602
5	PE6	Besouro de sinalização sonora	MINI_SPEAKER
6	V _{BAT}	Alimentação da bateria de backup	V _{BAT}
7	PC13	Notificação de que há uma ligação TCP aberta no Módulo WiFly	WIFLY_TCP_CONN_STAUTS
8	PC14	Ligação ao cristal do RTC (32,768kHz)	OSC32_IN
9	PC15	Ligação ao cristal do RTC (32,768kHz)	OSC32_OUT
10	V _{SS5}	Massa	GND
11	V _{DD5}	Alimentação +3,3V	VDD
12	PH0	Ligação ao cristal do sistema (8MHz)	OSC_IN
13	PH1	Ligação ao cristal do sistema (8MHz)	OSC_OUT
14	NRST	Reset de sistema	RESET
15	PC0	Pedido de interrupção do Módulo de áudio	Audio_DREQ
16	PC1	Pino do interface RMII – <i>Management Data Clock</i>	ETH_RMII_MDC
17	PC2	<i>Data Chip Select</i> do Módulo de áudio	Audio_xDCS
18	PC3	Pino do RMII -	ETH_RMII_TX_CLK
19	V _{DD12}	Alimentação de 3,3V	VDD
20	V _{SSA}	Massa do ADC	GND
21	V _{REF+}	Tensão de referência para o ADC	VREF+
22	V _{DDA}	Alimentação do sistema ADC	VDD
23	PA0	Pino a utilizar para acordar o μ C do modo de dormência mais profundo	WKUP
24	PA1	Pino do interface RMII – <i>Reference Clock</i>	ETH_RMII_REF_CLK
25	PA2	Pino do RMII – <i>Management Data I/O</i>	ETH_RMII_MDIO
26	PA3	Medição da corrente da corrente consumida pelo módulo de controlo das tampas	I_mTampas
27	V _{SS4}	Massa	GND
28	V _{DD4}	Alimentação de 3,3V	VDD
29	PA4	Pino não utilizado (disponível na PCI)	PA601
30	PA5	Controlo da linha 0 (Módulo de controlo das tampas)	TAMPA_L0
31	PA6	Medição da tensão V _{USB}	ADC12_IN6
32	PA7	Pino do interface RMII – <i>Carrier Sense/Receive Data Valid</i>	ETH_RMII_CVS_DV
33	PC4	Pino do interface RMII – <i>Receive Data 0</i>	ETH_RMII_RXD0
34	PC5	Pino do interface RMII – <i>Receive Data 1</i>	ETH_RMII_RXD1
35	PB0	Medição da tensão da bateria (V _{BAT})	ADC12_IN8
36	PB1	Controlo da linha 1 (Módulo de controlo das tampas)	TAMPA_L1
37	PB2	Controlo da linha 2 (Módulo de controlo das tampas)	TAMPA_L2
38	PE7	Controlo da linha 3 (Módulo de controlo das tampas)	TAMPA_L3
39	PE8	Controlo da linha 4 (Módulo de controlo das tampas)	TAMPA_L4
40	PE9	Controlo da coluna 0 (Módulo de controlo das tampas)	TAMPA_C0
41	PE10	Controlo da coluna 1 (Módulo de controlo das tampas)	TAMPA_C1

Nº do pino	Nome do pino	Função do pino	Nome da ligação
42	PE11	Controlo da coluna 2 (Módulo de controlo das tampas)	TAMPA_C2
43	PE12	Controlo da coluna 3 (Módulo de controlo das tampas)	TAMPA_C3
44	PE13	Controlo da coluna 4 (Módulo de controlo das tampas)	TAMPA_C4
45	PE14	Controlo da coluna 5 (Módulo de controlo das tampas)	TAMPA_C5
46	PE15	Controlo da coluna 6 (Módulo de controlo das tampas)	TAMPA_C6
47	PB10	I ² S – Linha de Relógio para o Módulo de áudio	I2S2_CK
48	PB11	Pino do interface RMII – <i>Transmit Enable</i>	ETH_RMII_TX_EN
49	V _{CAP_1}	Pino de ligação de condensador de desacoplamento do regulador de tensão interno	V _{CAP_1}
50	V _{DD1}	Alimentação de 3,3V	VDD
51	PB12	Pino do interface RMII – <i>Transmit Data 0</i>	ETH_RMII_TXD0
52	PB13	Pino do interface RMII – <i>Transmit Data 1</i>	ETH_RMII_TXD1
53	PB14	I ² S – Pino extra para com. <i>full duplex</i> – mapeado no MISO-SPI	I2S2ext_SD
54	PB15	I ² S – Pino <i>Serial Data</i> – mapeado no MOSI-SPI	I2S2_SD
55	PD8	Pino de TX da USART3	WiFly_RX
56	PD9	Pino de RX da USART3	WiFly_TX
57	PD10	GPIO disponível no conector do módulo Bluetooth	BT_GPIO1
58	PD11	Pino de CTS da USART3	WiFly_CTS
59	PD12	Pino de RTS da USART3	WiFly_RTS
60	PD13	Pino de Reset do Módulo WiFly	WIFLY_RST
61	PD14	Pino de indicação do estado da ligação à rede Wi-Fi	WIFLY_ASS_STATU S
62	PD15	Pino de ativação de uma ligação TCP utilizando a rede Wi-Fi	WIFLY_TCP_CONN_ TRIG
63	PC6	I ² S – Master Clock	I2S2_MCK
64	PC7	Pino não utilizado (disponível na PCI)	PC601
65	PC8	Pino D0 do interface SDIO (Ligação Micro-SD)	SDIO_D0
66	PC9	Pino D1 do interface SDIO (Ligação Micro-SD)	SDIO_D1
67	PA8	GPIO disponível no conector do módulo <i>Bluetooth</i>	BT_GPIO0
68	PA9	Pino de TX da USART1	BT_RX
69	PA10	Pino de RX da USART1	BT_TX
70	PA11	Pino de ligação à porta <i>micro-USB</i>	USB_DM
71	PA12	Pino de ligação à porta <i>micro-USB</i>	USB_DP
72	PA13	Pino JTMS (Interface SWD)	JTMS
73	V _{CAP_2}	Pino de ligação de condensador de desacoplamento do regulador de tensão interna	V _{CAP_2}
74	V _{SS2}	Massa	VSS
75	V _{DD2}	Alimentação 3,3V	VDD
76	PA14	Pino JTCK (Interface SWD)	JTCK
77	PA15	Pino SD1 do interface SDIO (Ligação Micro-SD)	SDIO_CD1
78	PC10	Pino D2 do interface SDIO (Ligação Micro-SD)	SDIO_D2
79	PC11	Pino D3 do interface SDIO (Ligação Micro-SD)	SDIO_D3
80	PC12	Pino CK do interface SDIO (Ligação Micro-SD)	SDIO_CK
81	PD0	GPIO disponível no conector do módulo GSM	GSM_GPIO0
82	PD1	GPIO disponível no conector do módulo GSM	GSM_GPIO1
83	PD2	Pino CMD do interface SDIO (Ligação Micro-SD)	SDIO_CMD
84	PD3	Pino de CTS da USART2	GSM_CTS
85	PD4	Pino de RTS da USART2	GSM_RTS
86	PD5	Pino de TX da USART2	GSM_RX

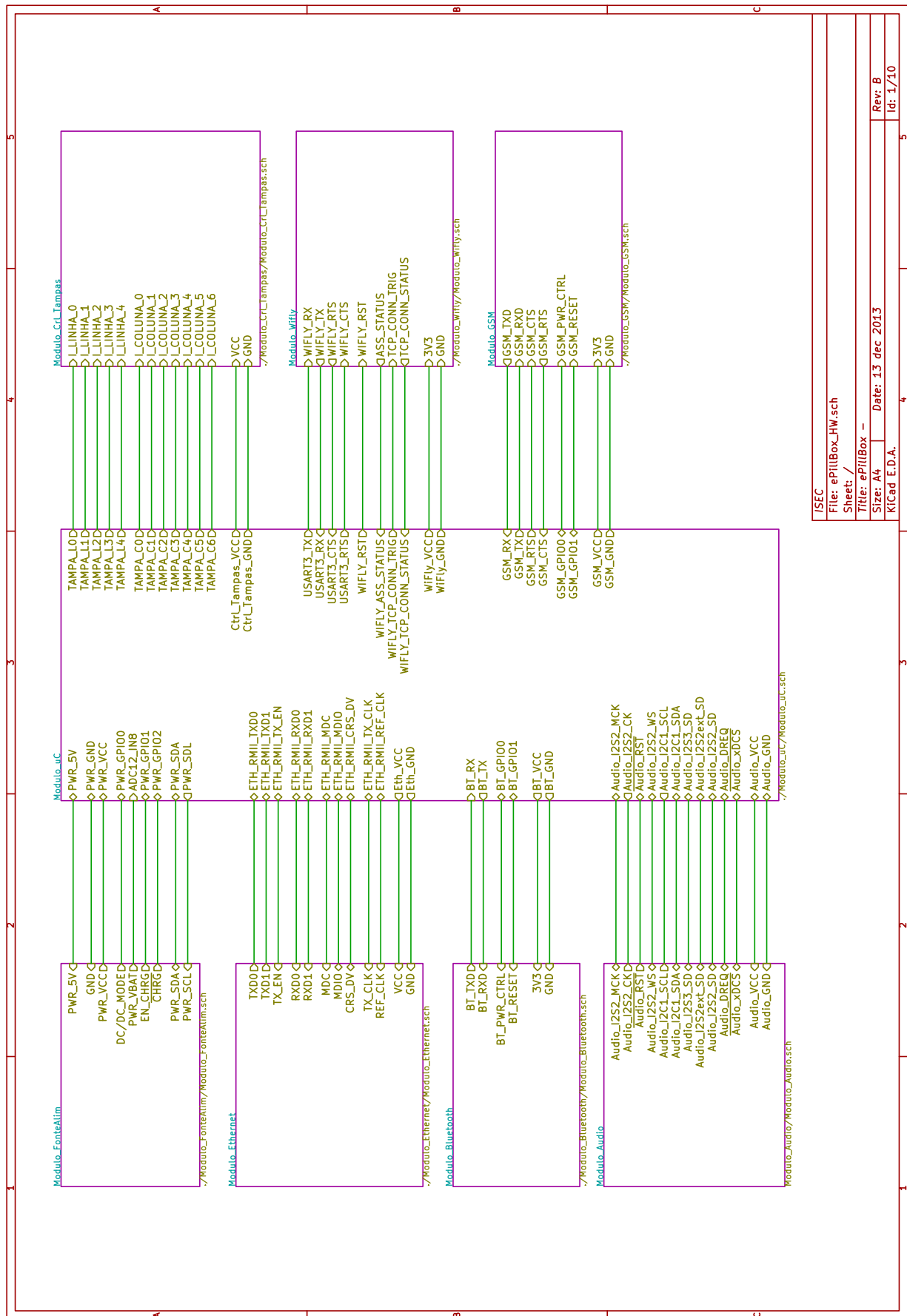
Nº do pino	Nome do pino	Função do pino	Nome da ligação
87	PD6	Pino de RX da USART2	GSM_TX
88	PD7	Pino não utilizado (disponível na PCI)	PD601
89	PB3	Pino JSWO (Interface SWD)	T_SWO
90	PB4	Reset do Módulo de áudio	Audio_RST
91	PB5	Pino não utilizado (disponível na PCI)	PB601
92	PB6	Pino de Relógio do Barramento I ² C	I2C1_SCL
93	PB7	Pino de dados do Barramento I ² C	I2C1_SDA
94	BOOT0	Configuração do <i>Boot</i> do μ C	BOOT0
95	PB8	Reset do CI de expansão de portas (MCP23008)	IOE1_RST
96	PB9	I ² S – Word Select – mapeado no pino NSS do SPI	I2S2_WS
97	PE0	Entrada/Saída digital Nº0	PE0
98	PE1	Entrada/Saída digital Nº1	PE1
99	V _{SS3}	Massa	VSS
100	V _{DD3}	Alimentação 3,3V	VDD

Anexo B - Esquemas eletrônicos do projeto

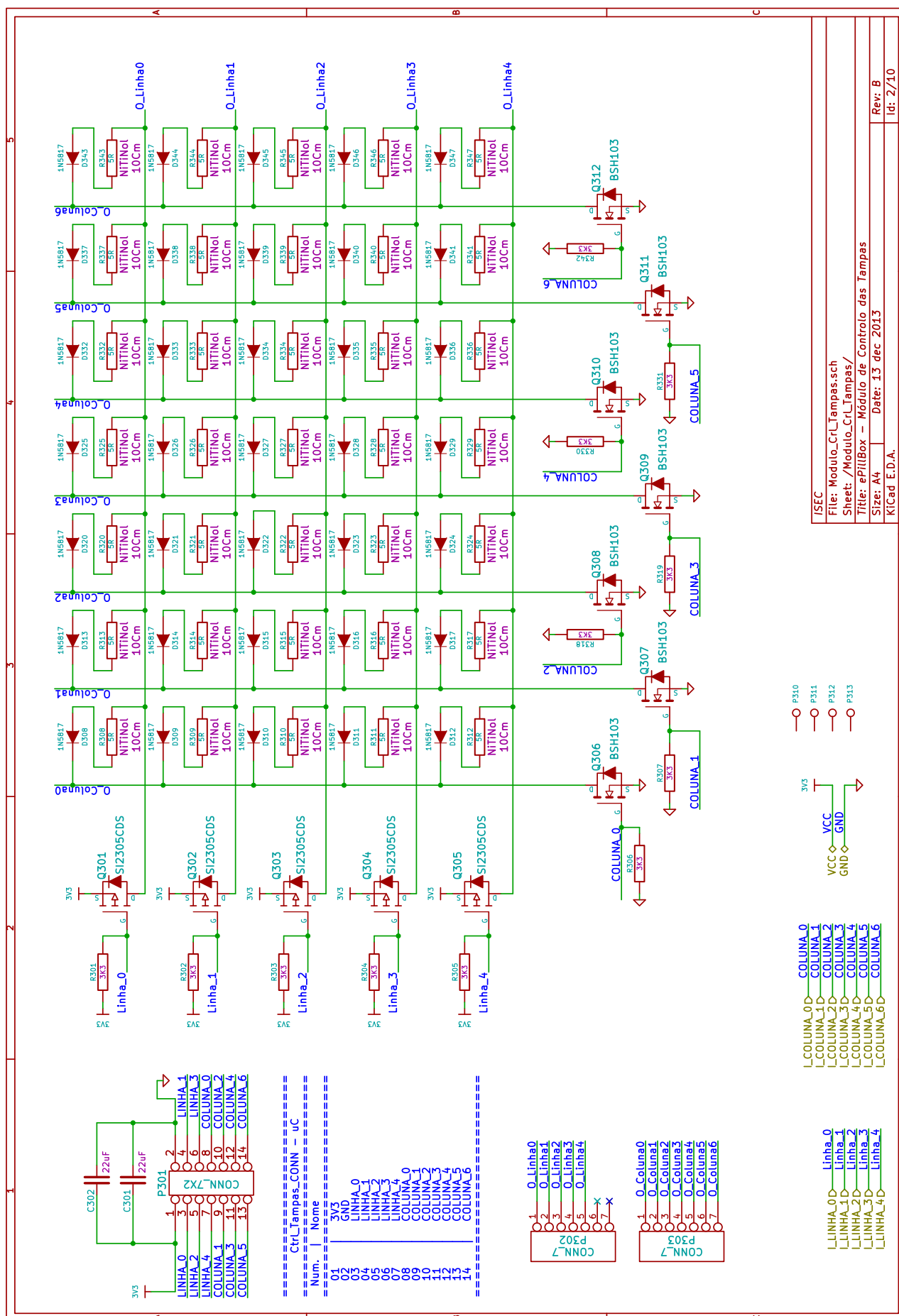
Índice

B.1 – Interligação entre Módulos	B-1
B.2 – Esquema do módulo de atuação das tampas	B-2
B.3 – Esquema do módulo <i>Bluetooth</i>	B-3
B.4 – Esquema do módulo do μC	B-4
B.5 – Esquema do módulo GSM	B-6
B.6 – Esquema do módulo da fonte de alimentação	B-7
B.7 – Esquema do módulo <i>Ethernet</i>	B-8
B.8 – Esquema do módulo Wi-Fi	B-9

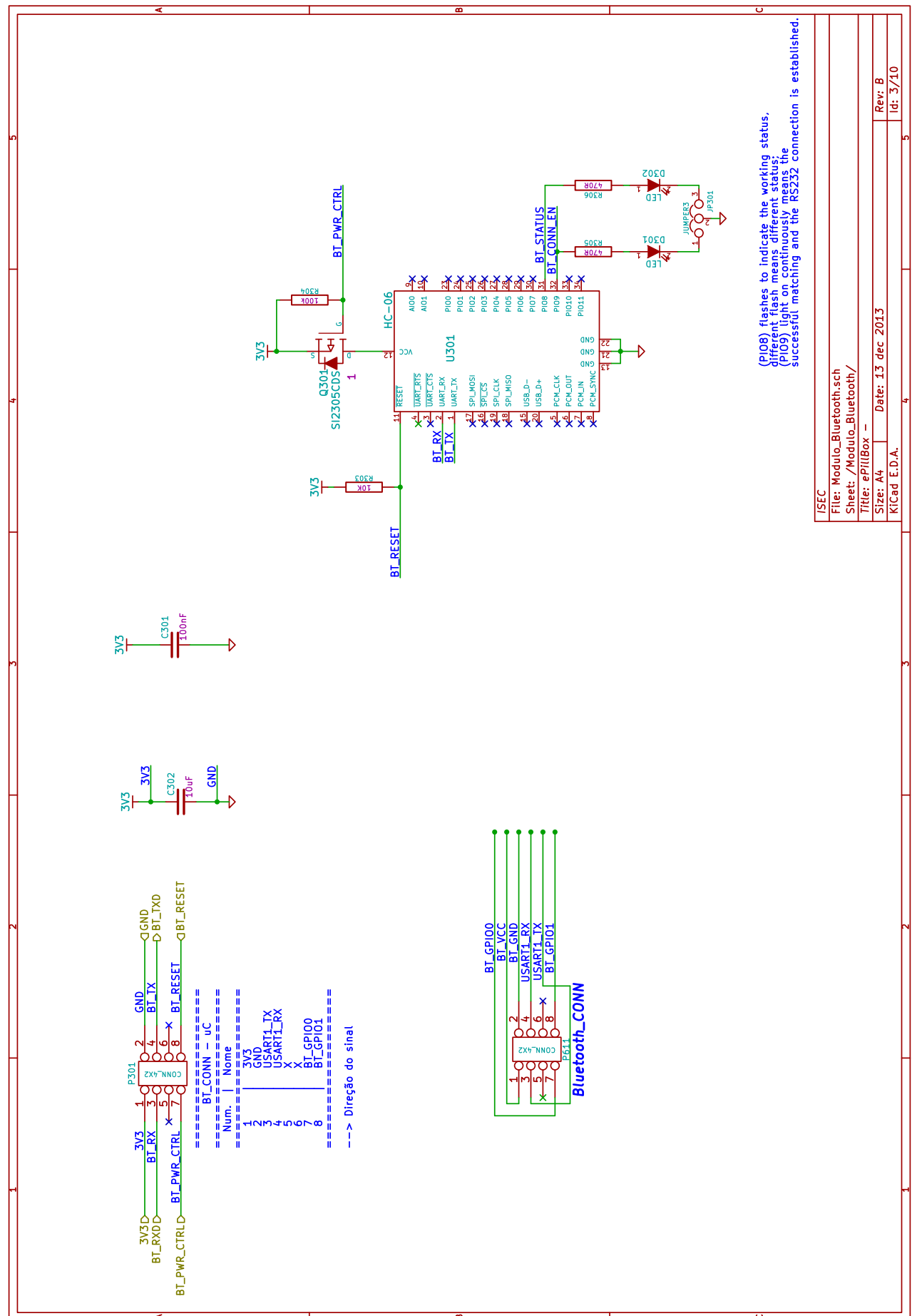
B.1 - Interligação entre Módulos



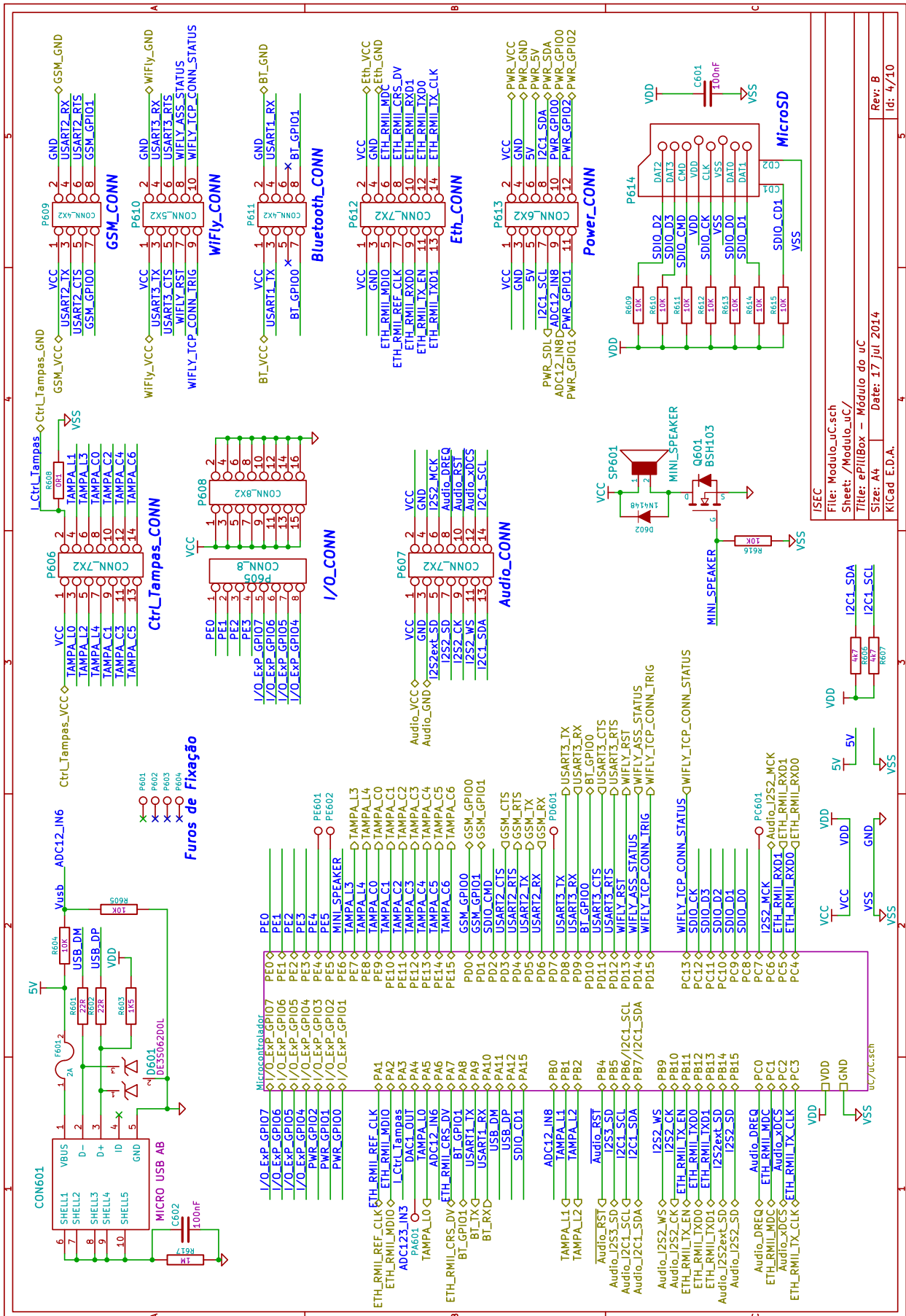
ISEC
 File: ePillBox_HW.sch
 Sheet: /
 Title: ePillBox -
 Size: A4 Date: 13 dec 2013 Rev: B
 KiCad E.D.A. Id: 1/10

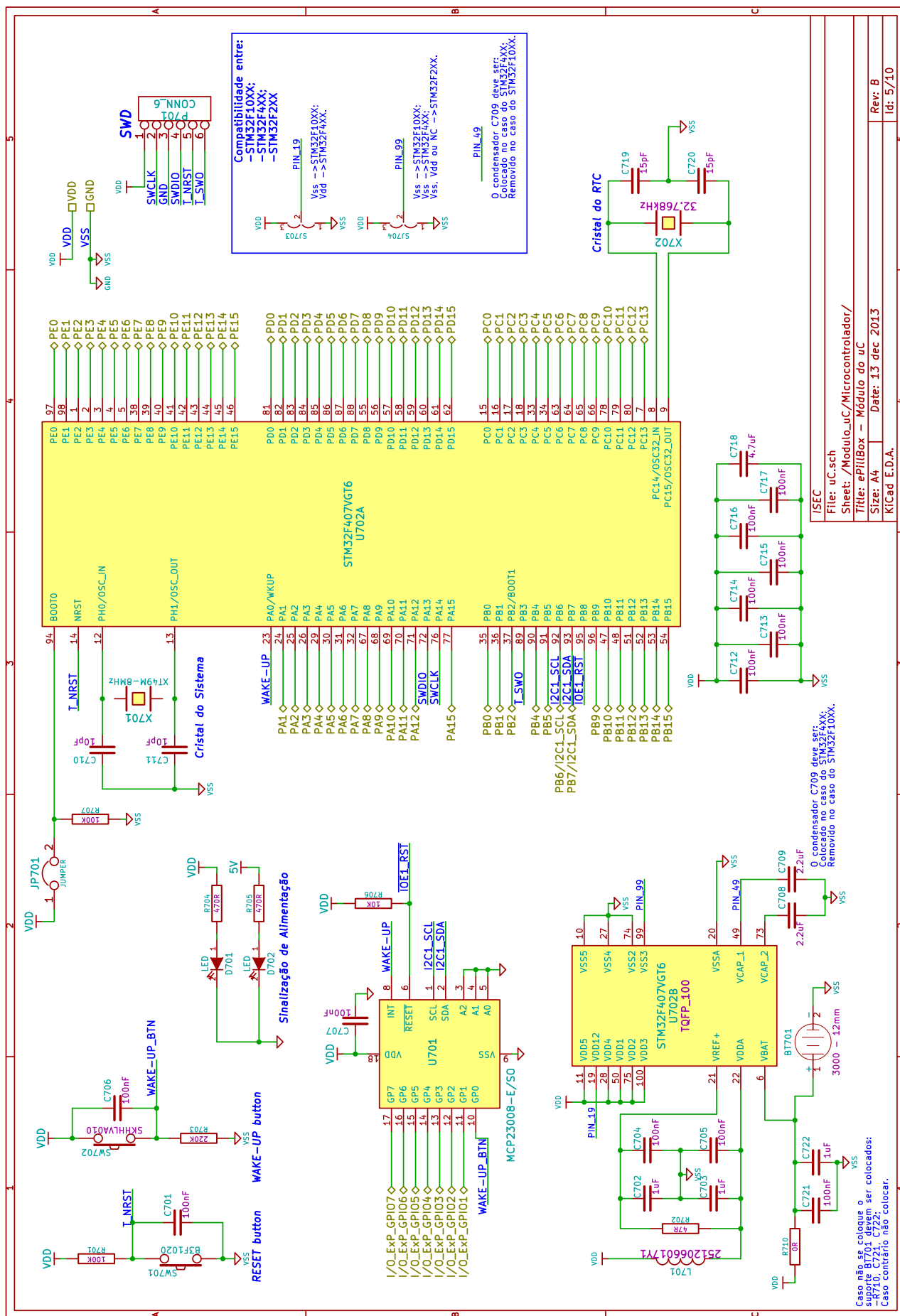


B.3 - Esquema do módulo *Bluetooth*



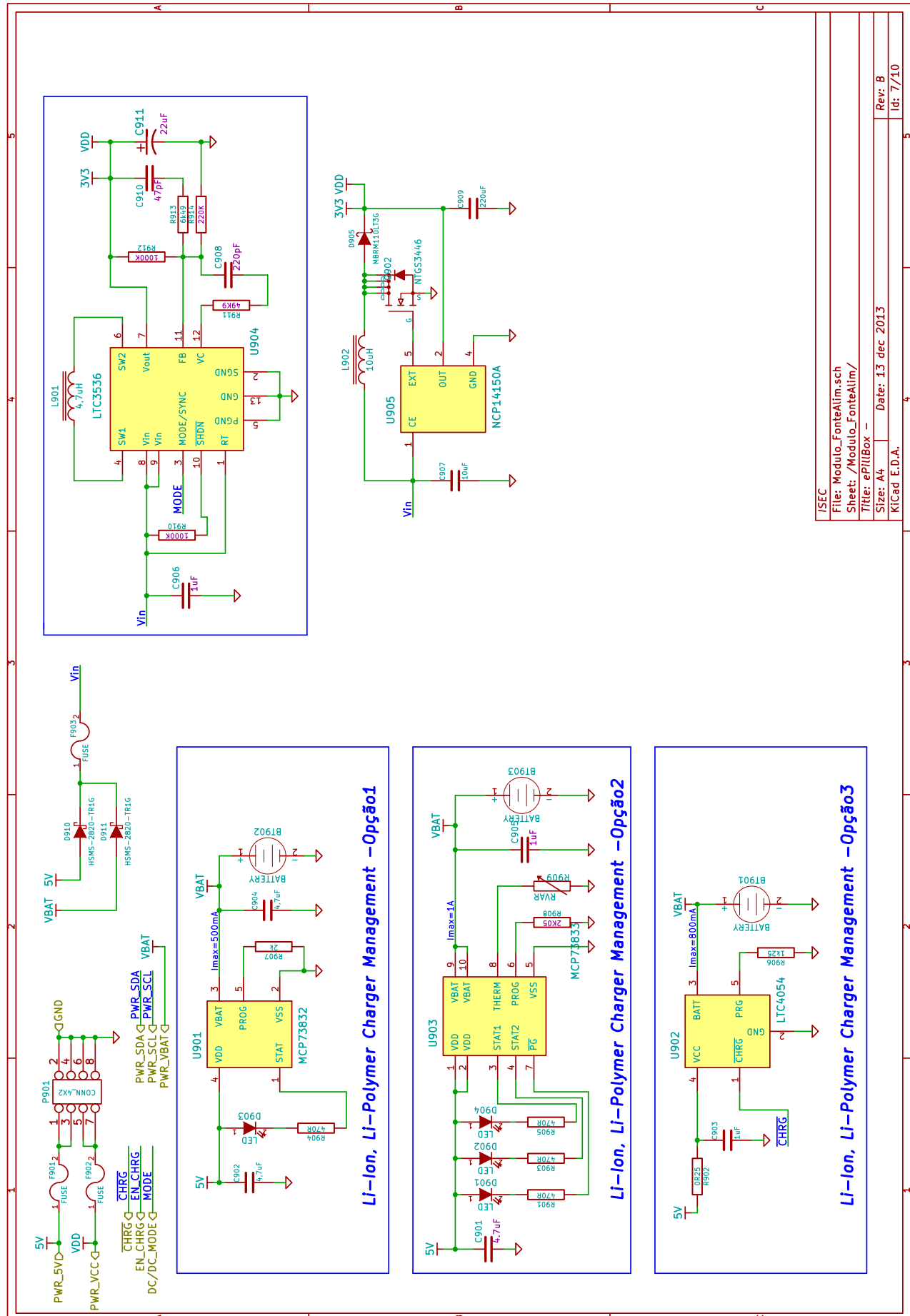
B.4 - Esquema do módulo do μC



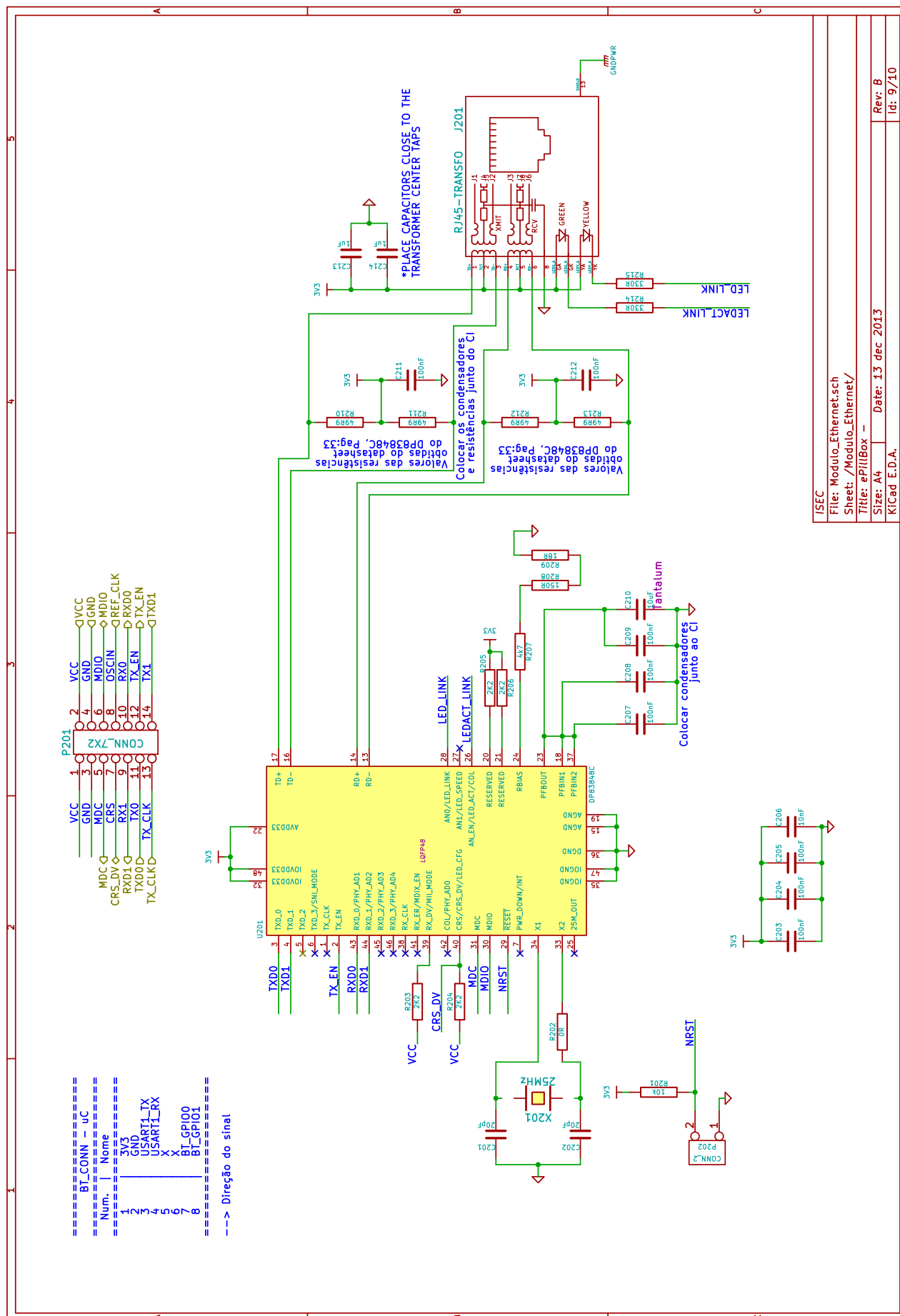




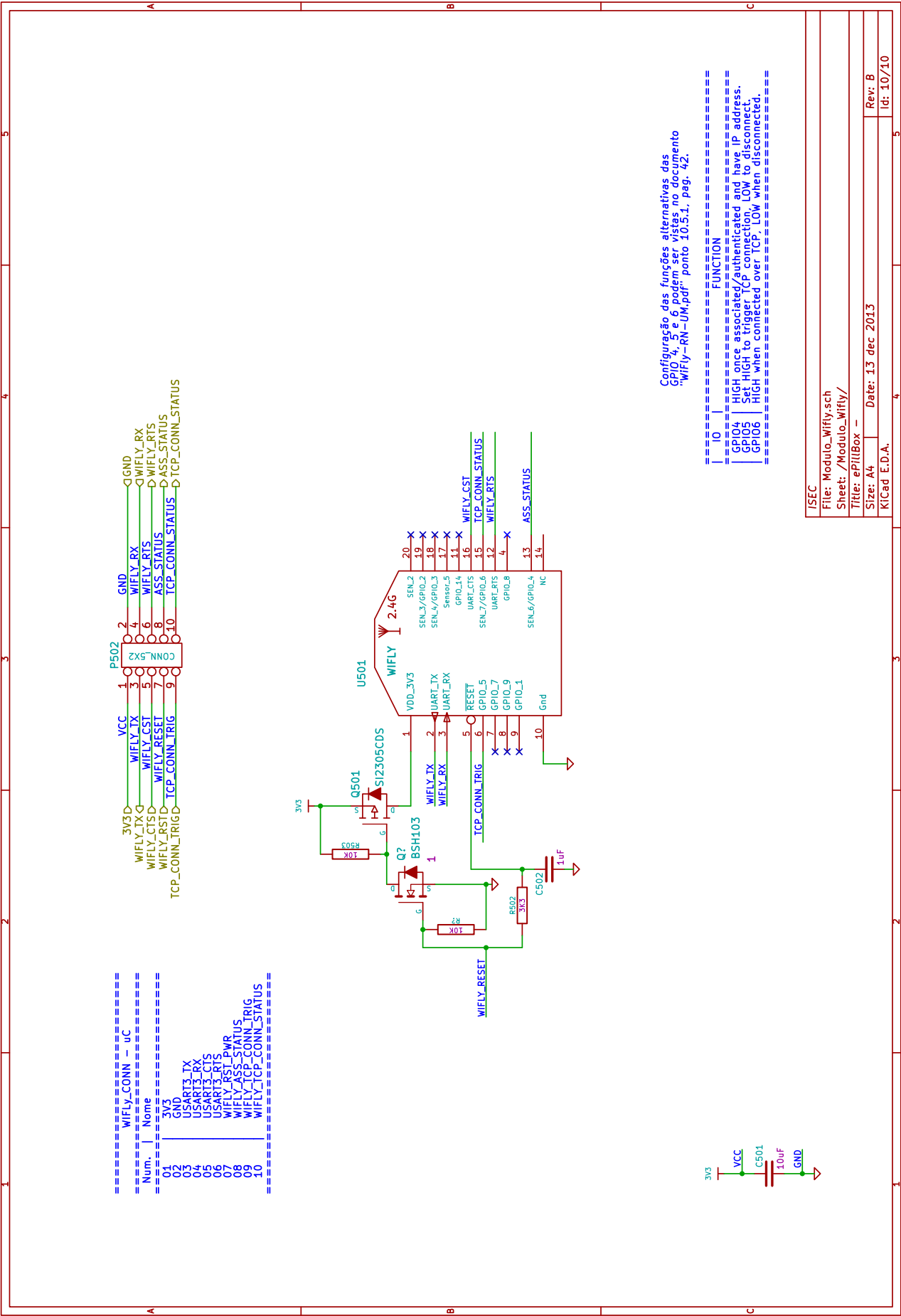
B.6 - Esquema do módulo da fonte de alimentação



B.7 - Esquema do módulo Ethernet



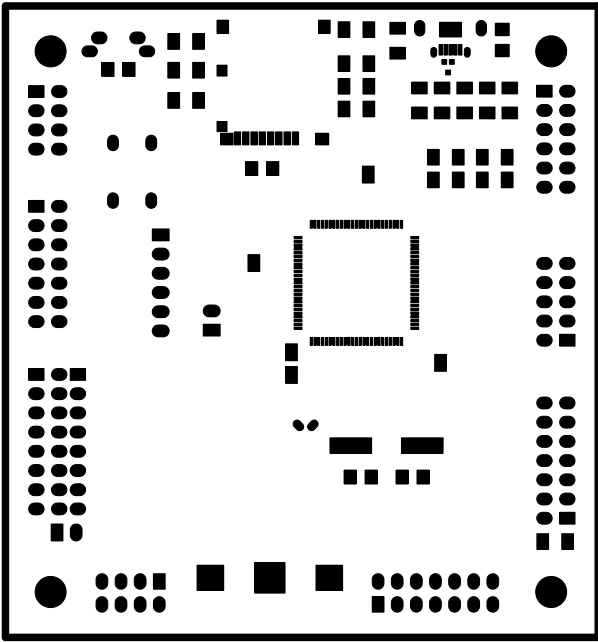
B.8 - Esquema do módulo Wi-Fi



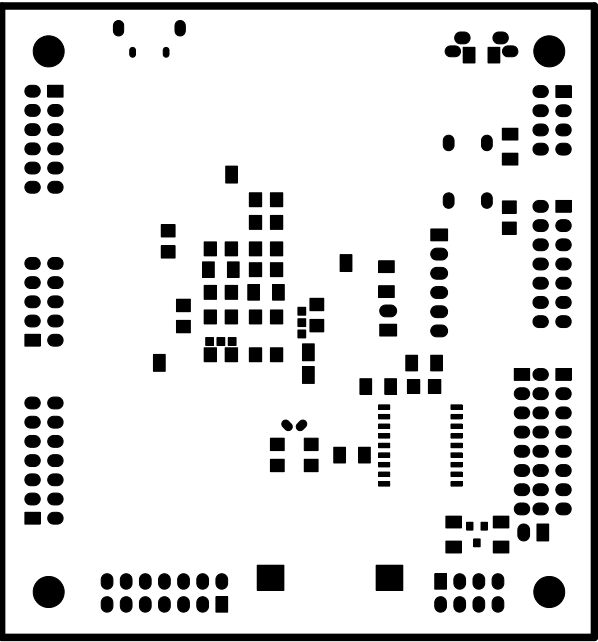
Anexo C - Desenho das Placas de Circuito Impresso

Índice

C.1 - Placa do Módulo do μC	C-1
C.2 - Placa do Módulo de controlo das tampas	C-3
C.3 - Placa do Módulo da fonte de alimentação	C-4
C.4 - Placa do Módulo de bluetooth	C-5

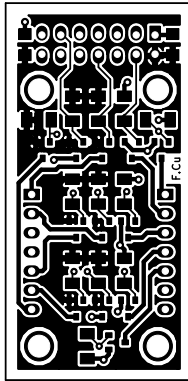


Mascara de solda superior

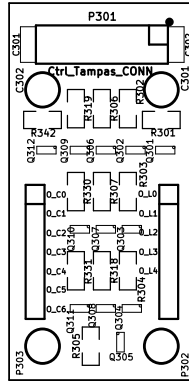


Mascara de solda inferior

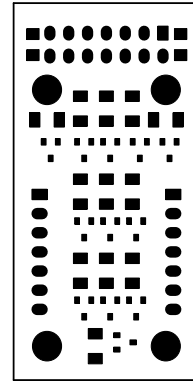
C.2 - Placa do Módulo de controlo das tampas



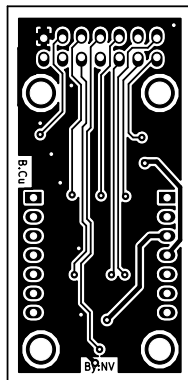
Camada de Cobre superior



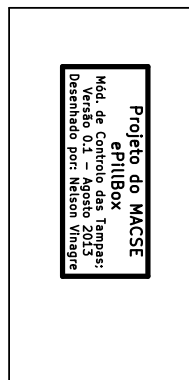
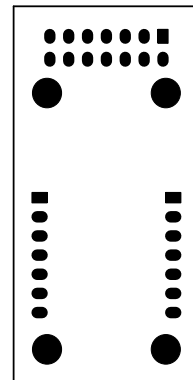
Camada *Silk Screen* superior

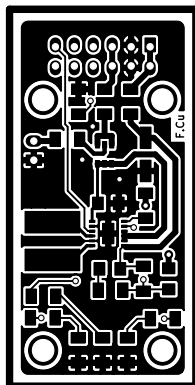


Camada de *Silk Screen*
superior

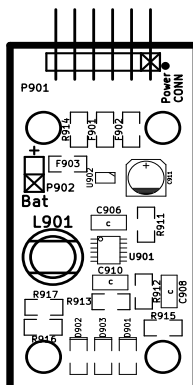
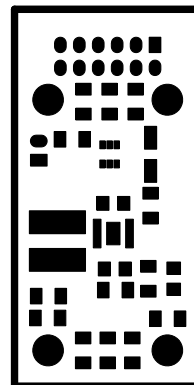
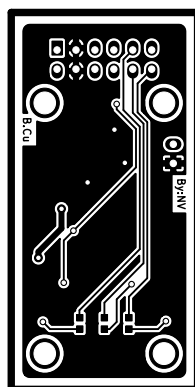


Camada de Cobre Inferior

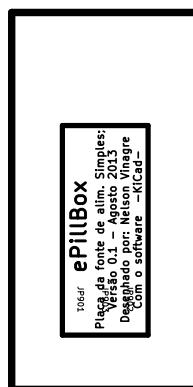
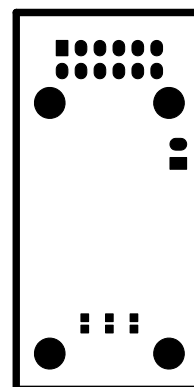
Camada *Silk Screen* InferiorCamada de *Silk Screen* Inferior

C.3 - Placa do Módulo da fonte de alimentação

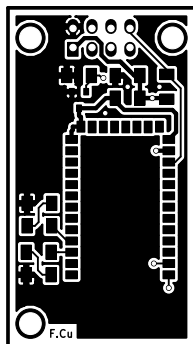
Camada de Cobre superior

Camada *Silk Screen* superiorCamada de *Silk Screen* superior

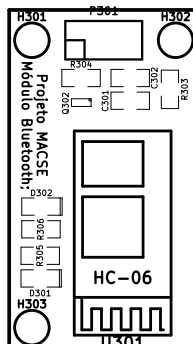
Camada de Cobre Inferior

Camada *Silk Screen* InferiorCamada de *Silk Screen* Inferior

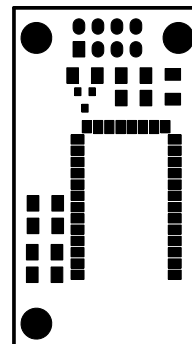
C.4 - Placa do Módulo de bluetooth



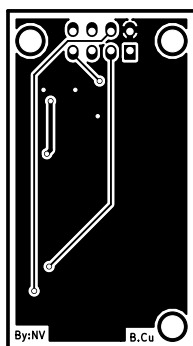
Camada de Cobre superior



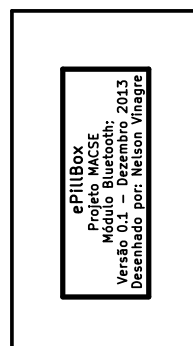
Camada *Silk Screen* superior



Camada de *Silk Screen* superior



Camada de Cobre Inferior



Camada *Silk Screen* Inferior



Camada de *Silk Screen* Inferior

Anexo D - Listagem e descrição das funções da API do servidor web

Índice

D.1 - Listagem de procedimentos da API	D-1
Tabela D.1 – Procedimento ‘ <i>Set_deviceRegistration</i> ’	D-1
Tabela D.2 – Procedimento ‘ <i>update_deviceRegistration</i> ’	D-1
Tabela D.3 – Procedimento ‘ <i>get_doseTable</i> ’	D-2
Tabela D.4 – Procedimento ‘ <i>update_doseTable</i> ’	D-2
Tabela D.5 – Procedimento ‘ <i>get_generalConfig</i> ’	D-2
Tabela D.6 – Procedimento ‘ <i>get_dateTime</i> ’	D-3
Tabela D.7 – Procedimento ‘ <i>report_PanicEvent</i> ’	D-3

D.1 - Listagem de procedimentos da API

Os procedimentos disponíveis na API do servidor web são os mínimos para permitir a troca de informação entre o dispositivo e o servidor. Os procedimentos implementados são os seguintes:

- **Set_deviceRegistration** – Regista o dispositivo e as suas características na base de dados (BD);
- **update_deviceRegistration** – Atualiza o registo do dispositivo e as suas características na BD;
- **get_doseTable** – Pede ao servidor a nova tabela de medicamentos;
- **update_doseTable** – Atualiza a tabela das tomas com as tomas já efetuadas;
- **get_generalConfig** – Pede ao servidor as configurações gerais para o dispositivo;
- **get_dateTime** – Pede a data e hora do servidor;
- **report_PanicEvent** – Reporta ao servidor que o botão de pânico foi pressionado;

Nas tabelas Tabela D.1D.1 a D.7 estão presentes mais detalhes sobre as respetivas funções.

Tabela D.1 – Procedimento 'Set_deviceRegistration'.

Nome da Função	Set_deviceRegistration
Parâmetros de entrada	Os parâmetros de entrada são: <ul style="list-style-type: none"> • UID: Número único de identificação do dispositivo; • Timestamp: da versão do <i>firmware</i>; • nDoses: Número de compartimentos do dispositivo;
Parâmetros de saída	O resultado pode ser: <ul style="list-style-type: none"> • Sucesso: e a inserção na tabela for bem-sucedida; • Erro: se falhar a inserção na tabela ou se o dispositivo estiver já registado;
Pré-requisitos	Este procedimento só pode ser chamado uma única vez antes de efetuar qualquer tipo de comunicação com o servidor;
Descrição	Este procedimento serve para o dispositivo se identificar ao servidor. A partir desse momento o servidor "sabe" da existência do equipamento e as suas características.

Tabela D.2 – Procedimento 'update_deviceRegistration'.

Nome da Função	update_deviceRegistration
Parâmetros de entrada	Os parâmetros de entrada são: <ul style="list-style-type: none"> • UID: Número único de identificação do dispositivo; • Timestamp: da versão do <i>firmware</i>; • nDoses: Número de compartimentos do dispositivo;
Parâmetros de saída	O resultado pode ser: <ul style="list-style-type: none"> • Sucesso: e a inserção na tabela for bem-sucedida; • Erro: se falhar a inserção na tabela ou se o dispositivo ainda não estiver registado;
Pré-requisitos	Este procedimento só pode ser chamado depois do dispositivo já se ter identificado ao dispositivo;
Descrição	Este procedimento serve para o dispositivo atualizar a sua identificação no servidor.

Tabela D.3 – Procedimento 'get_doseTable'.

Nome da Função	get_doseTable
Parâmetros de entrada	Os parâmetros de entrada são: <ul style="list-style-type: none"> • UID: Número único de identificação do dispositivo;
Parâmetros de saída	O resultado pode ser: <ul style="list-style-type: none"> • Erro: se falhar na obtenção da tabela da base de dados ou se o dispositivo estiver já registado; • Sucesso: devolve uma tabela com a lista dos horários das tomas;
Pré-requisitos	Este procedimento só pode ser chamado depois do dispositivo estar registado;
Descrição	Este procedimento serve para o dispositivo obter a lista de tomas disponível para o dispositivo;

Tabela D.4 – Procedimento 'update_doseTable'.

Nome da Função	update_doseTable
Parâmetros de entrada	Os parâmetros de entrada são: <ul style="list-style-type: none"> • UID: Número único de identificação do dispositivo; • Tabela das tomas: Cópia da tabela das tomas guardada em memória;
Parâmetros de saída	O resultado pode ser: <ul style="list-style-type: none"> • Erro: se falhar a atualização tabela na BD ou se o dispositivo estiver já registado; • Sucesso: caso a atualização tabela na BD seja feita com sucesso;
Pré-requisitos	Este procedimento só pode ser chamado depois do dispositivo estar registado;
Descrição	Este procedimento serve para o dispositivo atualizar a tabela das tomas no servidor. Caso o dispositivo reporte uma dose não tomada o procedimento tratará de despoletar os mecanismos internos do servidor para tratar esse não cumprimento.

Tabela D.5 – Procedimento 'get_generalConfig'.

Nome da Função	get_generalConfig
Parâmetros de entrada	Os parâmetros de entrada são: <ul style="list-style-type: none"> • UID: Número único de identificação do dispositivo;
Parâmetros de saída	O resultado pode ser: <ul style="list-style-type: none"> • Erro: se falhar a obtenção da informação da base de dados ou se o dispositivo estiver já registado; • Sucesso: devolve uma tabela com a lista dos parâmetros pedidos;
Pré-requisitos	Este procedimento só pode ser chamado depois do dispositivo estar registado;
Descrição	Este procedimento serve para o dispositivo obter a lista de configurações gerais para o dispositivo;

Tabela D.6 – Procedimento 'get_dateTime'.

Nome da Função	<i>get_dateTime</i>
Parâmetros de entrada	Os parâmetros de entrada são: <ul style="list-style-type: none"> • UID: Número único de identificação do dispositivo;
Parâmetros de saída	O resultado pode ser: <ul style="list-style-type: none"> • Erro: se o dispositivo não estiver registrado; • Sucesso: retorna o hora e data do servidor no formato Ano, Mês, Dia, Hora, Minuto, Segundo, utilizando 2 dígitos para cada valor. <ul style="list-style-type: none"> ○ Exemplo: O valor recebido '131201112500' é equivalente a: 2013/12/01 – 11:25:00;
Pré-requisitos	Este procedimento só pode ser chamado depois do dispositivo estar registrado;
Descrição	Este procedimento serve para o dispositivo reportar ao servidor que o botão de pânico foi pressionado;

Tabela D.7 – Procedimento 'report_PanicEvent'.

Nome da Função	<i>report_PanicEvent</i>
Parâmetros de entrada	Os parâmetros de entrada são: <ul style="list-style-type: none"> • UID: Número único de identificação do dispositivo;
Parâmetros de saída	O resultado pode ser: <ul style="list-style-type: none"> • Erro: se falhar a obtenção da informação da base de dados ou se o dispositivo não estiver registrado; • Sucesso: caso todo ocorra dentro do esperado;
Pré-requisitos	Este procedimento só pode ser chamado depois do dispositivo estar registrado;
Descrição	Este procedimento serve para o dispositivo reportar ao servidor que o botão de pânico foi pressionado;

Anexo E - Parametrização do módulo Wi-Fi (RN-171-XV WiFly)

Índice

E.1 - Introdução	E-1
E.2 - Esquemático do módulo	E-1
E.3 - Listagem de parâmetros alterados	E-1
E.4 - Listagem de comandos para alterar os parâmetros	E-2
E.5 - Referências	E-3

E.1 - Introdução

O módulo RN-171-XV é uma solução de Wi-Fi simples e fácil de utilizar tanto em projetos novos como antigos. O módulo oferece para além do rádio 802.11/b/g, um processador de 32-bit, *stack* TCP/IP, Relógio de tempo real, unidade de gestão de energia, vários pinos de entrada/saída digitais, entradas analógicas, etc. O *firmware* do dispositivo faz todo o trabalho “pesado” da parte do Wi-Fi, permitindo assim a utilização de um processador mais simples e a redução da complexidade do *firmware* do sistema hospedeiro.

Para o restante documento, a versão do *firmware* considerada será a versão 4.0 ou superior. Caso o módulo seja adquirido com uma versão inferior deverá ser realizada uma atualização para a versão 4 ou superior. Para instruções de como realizar a atualização do *firmware* pode ser encontrada no ponto 4.5 – “Upgrading Firmware Via Ftp”, pág. 83 de [2].

E.2 - Esquemático do módulo

O módulo tem o aspeto da figura e.1 a) e nela pode ver-se a indicação do número dos pinos. A figura e.1 b) faz a correspondência entre o número do pino e o respetivo nome.

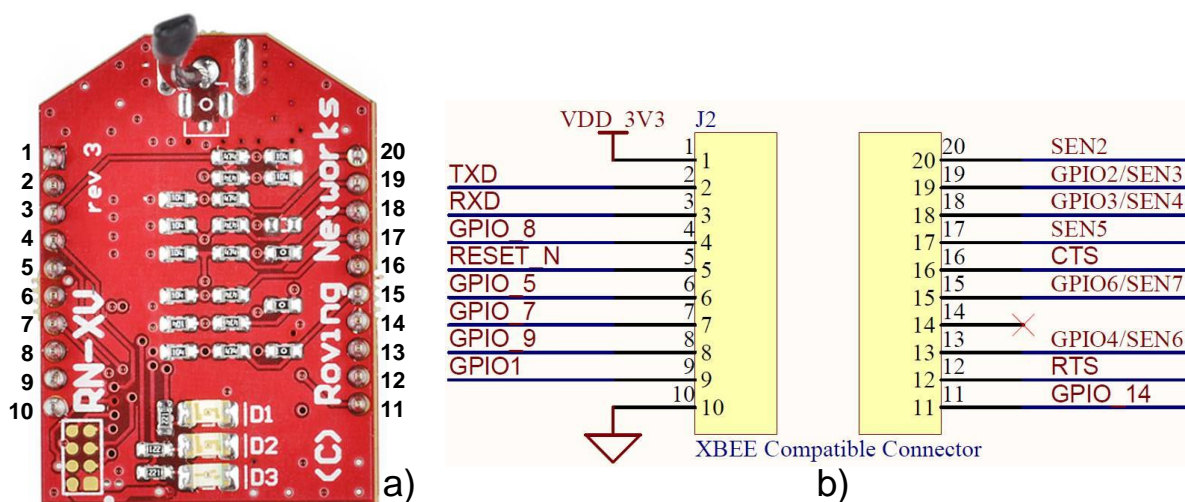


Figura E.1 – (a)Imagem do módulo RN-171-XV; (b)Esquema de corresp. entre o número do pino e o nome[1];

O esquema interno do módulo está presente no datasheet [1].

E.3 - Listagem de parâmetros alterados

O *firmware* do módulo possui vários parâmetros que definem a forma como o módulo opera. A lista dos parâmetros e respetivos valores por omissão estão presentes em [2]. Na tabela seguinte está a lista de todos os parâmetros que foram alterados para permitir o correto funcionamento do módulo com o microcontrolador de sistema.

Figura E.2 – Listagem de parâmetros alterados

Parâmetros	Valores por defeito	Novo Valor	Descrição
Com	Flush timer	10ms	250ms
	Remote	*HELLO*	Remove a string de “cumprimento” do servidor

Parâmetros	Valores por defeito	Novo Valor	Descrição	
IP			remoto.	
	Protocol	2 (TCP server and client)	0x12	Ativa o modo de HTTP (0x10) e modo TCP (0x2).
	Sleep timer	0	5s	Coloca o módulo em modo de hibernação passado 5 segundo de terminar uma comunicação.
System	Trigger	0x1	0x2	Força o módulo a abandonar o modo de hibernação ao se colocar o pino CTS a 1 (que internamente está ligado ao pino do sensor 1)
	IOfunc	0	0x70	Ativa a função alternativa das GPIO 4 a 6 que têm as funcionalidades de indicação e controlo de algumas funcionalidades do módulo.
TIME SERVER	Enable	0	1	Ativa a utilização dum servidor sNTP para sincronismo do relógio interno do módulo.
	Zone	7 (Pacific time, USA)	0(GMT)	Parametrização do fuso horário do módulo.
UART	Baud rate	9600	115200	Altera a <i>Baud rate</i> de 9600bps para 115200bps
	Mode	0	1	Desativa o eco da UART
WLAN	Join	0	1	Tenta liga-se a uma rede Wifi especificada assim que o módulo é ativado.
OPT	device_id	WiFly-GSX	ePillBox-AP	

E.4 - Listagem de comandos para alterar os parâmetros

Para realizar esta alteração dos parâmetros são necessários alguns meios. Desde já é necessário:

- Um PC com uma porta USB livre;
- Um conversor USB para rs232 para níveis de tensão de 3,3V;
- *Software* que permita comunicar através duma porta série (exemplo: *Tera Term*, *Putty*, *RealTerm*, etc.)

Depois de ter o módulo RN-171-XV corretamente ligado ao conversor e abrir o *software* apropriado a sequência de procedimento é a seguinte:

- 1) No software escolher a porta série correta e configurá-la com os seguintes parâmetros: 9600bps, 8bits, sem paridade, 1stop bit, sem controlo de fluxo.
- 2) Estabelecer a ligação ao módulo;
- 3) Colocar o módulo em modo de comando. Para entrar em modo comando é necessário enviar a *string* '\$\$\$' (3 sinais de Dólar). Se tudo correr bem deverá aparecer no terminal a *string* 'CMD'.
- 4) Uma vez em modo de comando já é possível alterar os parâmetros definidos na tabela acima. Para os alterar é só correr os seguintes comandos.

Comando	Descrição
set uart mode 1	//Desativa o eco do módulo
set comm time 500	//Coloca o valor do "Flush timer" para 500ms
set comm size 1460	// coloca o buffer de entrada para 1460 bytes que é o tamanho máximo de um quadro Ethernet.
set comm remote 0	// remover a string que é enviada ao servidor remoto quando a porta é aberta.
set ip proto 18	//Configura o protocolo para HPPT e TCP
set sys sleep 5	//Configura o tempo para entrar em hibernação para 5s.
set sys trigger 2	//Configura o módulo a acordar ao colocar o pino CTS a 1;
set sys iofunc 0x70	//Ativa a função alternativa das GPIO 4-6
set time enable 1	// Activa o módulo para sempre que se ligar a uma rede Wifi atualizar o seu tempo através do sNTP.
set time zone 0	//Configura o fuso horário para GMT
set uart baud 115200	//Coloca a baud rate a funcionar a 115200bps
set wlan join 1	// Tenta liga-se a uma rede Wifi especificada assim que o módulo é ativado.
set opt device_id ePillBox-AP	// configura o nome do dispositivo para 'ePillBox-AP'
save	//Guarda a configuração
reboot	//Força o módulo a fazer o reboot e a carregar as configurações acabadas de fazer.

- 5) Depois desta tarefa executada com sucesso o módulo deixará de responder já que a velocidade de comunicação foi alterada. Torna-se assim necessário realizar a alteração da velocidade de comunicação para 115200bps.
- 6) Após realizar a alteração é possível verificar se o módulo está a operar corretamente tentando entrar em modo de comando. Caso se consiga, o módulo está corretamente parametrizado e pode ser colocado a funcionar no equipamento.

E.5 - Referências

- [1] Roving Networks, Data Sheet do "RN-171-XV 802.11 b/g Wireless LAN Module", versão 1.04 - 10/29/2012;
- [2] Roving Networks, "WiFly Command Reference, Advanced Features & Applications User's Guide", versão 1.2r - 30/04/2013;